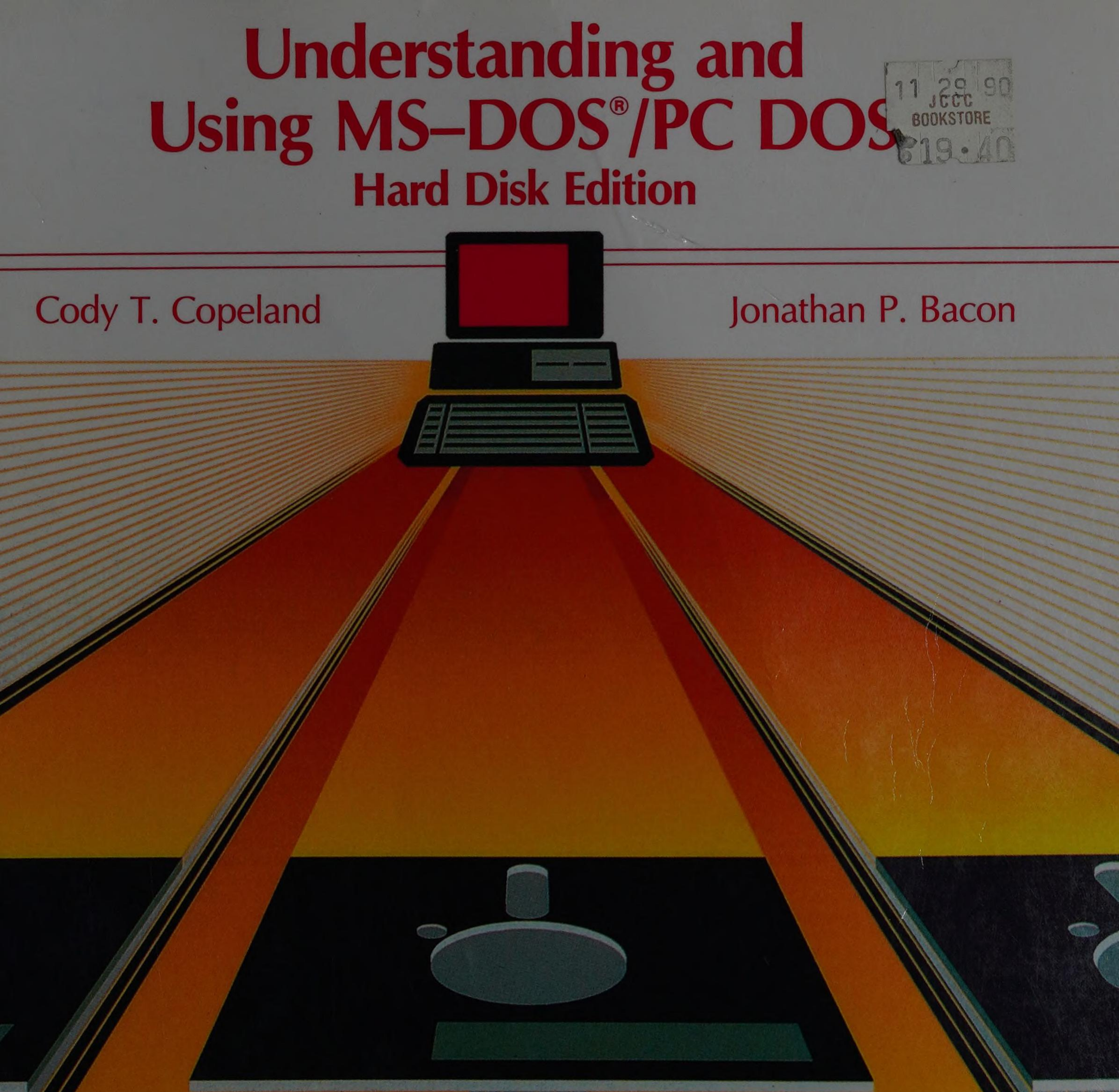
Understanding and Using MS-DOS®/PC DOS

Hard Disk Edition

11 29 90
JCCC
BOOKSTORE
$19.40

Cody T. Copeland

Jonathan P. Bacon



UNDERSTANDING AND USING MS-DOS®/PC DOS

Hard Disk Edition

Cody C. Copeland
Johnson County Community College

Jonathan P. Bacon
Johnson County Community College

West Publishing Company
St. Paul New York Los Angeles San Francisco

Cover Design: Bob Anderson, Computer Arts, Inc.

CREDITS:

Microsoft, Microsoft Word, MS-DOS, and MS-NET are registered trademarks of Microsoft Corporation, Redmond, WA.
3Com is a registered trademark of 3Com Corporation.
10-Net is a registered trademark of 10-NET Communications, a division of DCA, Dayton, OH.
Apple is a registered trademark of Apple Corporation, Cupertino, CA.
Burroughs is a registered trademark of Unisys Corporation, Detroit, MI.
Datacomm, Intercom 1000, and Intercom 1000+ are registered trademarks of Intercomputer Communications Corporation (ICC), Cincinnati, OH.
HP LaserJet Plus and HP LaserJet Series II are registered trademarks of Hewlett Packard Company.
IBM, IBM EGA, IBM Personal Computer, IBM PC, IBM Personal System/2, PS/2, IBM PC-NET, IBM PC/XT, IBM PC/AT, IBM PC-DOS, Operating System/2, OS/2, and Presentation Manager are registered trademarks of the International Business Machines Corporation.
Intel is a registered trademark of the Intel Corporation.
Novell and NetWare are registered trademarks of Rohm Corporations.
ROHM and Juniper II are registered trademarks of Rohm Corporation.
SmartWare II is a registered trademark of Informix Software, Inc., Lenexa, KS.

COPYRIGHT © 1989 by WEST PUBLISHING CO.
COPYRIGHT © 1990 by WEST PUBLISHING CO.
50 West Kellogg Boulevard
P.O. Box 64526
St. Paul, MN 55164-1003

All rights reserved

Printed in the United States of America

97 96 95 94 93 92 91 90 8 7 6 5 4 3 2 1 0

Library of Congress Cataloging-in-Publication Data

Copeland, Cody T.
Understanding and using MS-DOS/PC DOS / Cody T. Copeland, Jonathan P. Bacon -- 2nd ed.
p. cm. -- (The Microcomputing series)
ISBN 0-314-66573-0
1. MS-DOS (Computer operating system) 2. PC DOS (Computer operating system)
I. Bacon, Jonathan. II. Title. III. Series.
QA76.76.063C666 1990
005.446--dc20 89-37876
CIP

CONTENTS

About the Authors xvii
Preface xix
Who Should Use this Book? xix
Content Highlights xix
Approach to the Text xx
Computer Terms Dictionary xx
Versions of DOS xxi
Changes from the 1987 Edition xxi
Acknowledgments xxii

Part I: An Overview of IBM PC-DOS 1

Unit 1 An Introduction to Operating Systems 3

Objectives 3
Assignments
An Introduction to Operating Systems 4
Booting Your System and the Power-on Self-test 6
Getting Started By Loading DOS 7
Types of Personal Computers 7
The Importance of the Disk Operating System 10
Computer Language Syntax: The Order that Promotes Understanding 11
Issuing DOS Commands 13
Internal Versus External DOS Commands 14
Filenames, Extensions, and Drive Designators 15
The Filename: Naming Conventions 15
The Filename Extension 17
DOS Filename Extensions 18
The Drive Designation 18

The Default Drive or Logged-on Drive 19
The Path 19
The Complete Filename 20
Wildcard Characters 20
Disk Handling Guidelines 21
Basic Assumptions 26
A History of DOS 26
Review Questions 27
Documentation Research 28

Part II Fundamental DOS Commands 29

Unit 2 Disk Preparation Commands 31

Objectives 31
Important Commands 31
Assignments 32
Disk Preparation 32
Disk Construction 32
5¼-Inch Disk 32
3½-Inch Disk 33
Watch those Diskettes 33
Hard Disks 33
Tracks and Sectors 34
Loading DOS 3.30 34
Guided Activity: Change the System Prompt 35
Guided Activity: Change to the DOS Subdirectory 35
Format Command 36
Everyday FORMAT Command Switches 36
Guided Activity: Format a Non-system Diskette 37
Guided Activity: Format a Non-system Diskette with a Volume Label 39
Guided Activity: Format a System Diskette 41
Guided Activity: Format a System Diskette with a Volume Label 43
SYS Command 45
Guided Activity: Transfer the DOS system Files with SYS the Command 45
LABEL Command 47
Guided Activity: Add a Volume Label to a Diskette 47
DISKCOPY Command 49
Guided Activity: Make a Duplicate Disk with the DISKCOPY Command 49
DISKCOMP Command 51
Guided Activity: Compare Two Diskettes with the DISKCOPY Command 51
Command Review Dictionary 53
Review Questions 53
Unit Exercises 54
Documentation Research 54

Application A 55

Part 1: Formatting a Blank Diskette 55
Part 2: Creating a Label with the LABEL Command 55

Unit 3 Housekeeping Commands 57

Objectives 57
Important Commands 58
Assignments 58
Housekeeping Commands 58
CLS Command 59
Guided Activity: Clear the Screen with the CLS Command 59
PROMPT Command 60
Guided Activity: Use the PROMPT Command to Change Your System Prompt 60
Guided Activity: View the Directory with the System Prompt 62
DATE Command 63
Guided Activity: Use the DATE Command to Change the System Date 63
TIME Command 66
Guided Activity: Use the TIME Command to Change the System Time 66
DIR Command 67
Guided Activity: Use the DIR Command to Scan a File 67
Guided Activity: Use <CTRL-S> to Stop the Screen from Scrolling 68
Guided Activity: Use the DIR Command to Produce Horizontal Filenames 69
Guided Activity: Use the DIR Command with a Pause 69
CHKDSK Command 70
Guided Activity: Check a Disk Using the CHKDSK Command 70
Guided Activity: Check a Disk Using CHKDSK with the /V Parameter 71
Guided Activity: Check a Disk Using CHKDSK with the /F Parameter 72
Guided Activity: Check a Hard Disk Using CHKDSK with the /F Parameter 73
Command Review Dictionary 77
Review Questions 77
Unit Exercises 77
Documentation Research 78

Unit 4 File Management Commands 79

Objectives 79
Important Commands 79
Assignments 80
File Management Commands 80
Guided Activity: Start the Computer 80
COPY Command 81
Guided Activity: Create a File Using the COPY Command 81
Guided Activity: Copy a File to the Screen 84
Guided Activity: Copy a File to Another Drive 85
Guided Activity: Create Another File 86
Guided Activity: Copy a File with Wildcard Characters 88

Guided Activity: Copy a File to the Same Drive and Directory 89
Guided Activity: Copy More Files with Wildcards 90
Guided Activity: Copy with *.* 91
Guided Activity: Combine Two Files 92
Guided Activity: Copy a File to the Printer 94
COMP Command 95
Guided Activity: Compare Two Files with the COMP Command 96
Rename Command 97
Guided Activity: Rename a File 97
TYPE Command 100
Guided Activity: Output to Screen with the TYPE Command 101
Delete/ERASE Command 102
Guided Activity: Delete a Single File from the Current Drive 102
Guided Activity: Delete a Group of Files with Wildcard Characters 103
Command Review Dictionary 106
Review Questions 106
Unit Exercises 107

Application B 109

Part 1: Formatting a Blank Diskette 109
Part 2: Creating Text Files with COPY CON 109
Part 3: Combining Files 110
Part 4: Renaming of Files 110

Unit 5 Organizational Commands 111

Objectives 111
Important Commands 111
Assignments 112
Hard-Disk Management Commands 112
Guided Activity: Start the Computer and Make a Text File 112
MD Command 113
Guided Activity: Make Subdirectories on a diskette 114
CD Command 115
Guided Activity: Change Subdirectories on a Diskette 115
Guided Activity: Make Another Subdirectory 116
Guided Activity: Copy a File to a Subdirectory 117
PATH Command 119
Guided Activity: Name the Path 119
Guided Activity: Set a New Path 122
TREE Command 123
Guided Activity: View Subdirectories and Files with the TREE Command 123
RD Command 124
Guided Activity: Remove Subdirectories from a Diskette 125
Command Review Dictionary 128
Review Questions 128
Unit Exercises 129

Documentation Research 129

Application C 131

Part 1: Making Subdirectories 131
Part 2: Copying Files to Subdirectories 132
Part 3: Removing Subdirectories 132

Part III Intermediate DOS Commands 133

Unit 6 The Editing DOS Command 135

Objectives 135
Important Commands 135
Assignments 136
EDLIN: the Text Editor 136
Guided Activity: Starting EDLIN 136
Guided Activity: Use the INSERT Command 137
Guided Activity: Insert a Line into an Existing File 139
Guided Activity: List Lines 141
Guided Activity: Use the Delete Command 143
Guided Activity: End and Save an EDLIN Text File 144
Guided Activity: Output to the Screen with the TYPE Command 145
Guided Activity: Use the Q Command to Abort a File 146
Guided Activity: Create a New EDLIN File 147
More EDLIN 148
Guided Activity: Make an Office Help Screen 149
Guided Activity: Insert New Lines 150
Guided Activity: Use the EDIT Command 152
Guided Activity: Use the COPY Command 154
Guided Activity: Use the MOVE Command 156
Guided Activity: Use the SEARCH Command 159
Guided Activity: Use the SEARCH Command with ? 160
Guided Activity: Use the REPLACE Command 162
Guided Activity: Clean Up and Save the File 164
Command Review Dictionary 166
Review Questions 166
Unit Exercises 166
Documentation Research 166

Application D 167

Part 1: Format a Blank Diskette 167
Part 2: Creating Text Files Using EDLIN 167
Part 3: Using the TYPE Command 167

Unit 7 The Invisible DOS Command 169

Objectives 169
Important Commands 169
Assignments 170
The Invisible DOS Command 170
Guided Activity: Create the First BAT File 171
Guided Activity: Add Invisible Lines to the HELLO.BAT File 173
Guided Activity: Terminate a Batch File 175
Guided Activity: Stop the BAT File with the PAUSE Command 177
Guided Activity: Create an AUTOEXEC.BAT File 179
Guided Activity: Reboot the Computer with AUTOEXEC.BAT 181
Configuration System File 182
BUFFERS Command 182
FILES Command 183
Guided Activity: Create a CONFIG.SYS File with COPY CON 183
Command Review Dictionary 185
Review Questions 185
Unit Exercises 185
Documentation Research 185

Application E 187

Part 1: Creating Batch Files 187
Part 2: Using Batch Files to Copy Files to Subdirectories 187
Part 3: Batch Files for Removing Subdirectories 187

Unit 8 DOS Safety Commands 189

Objectives 189
Important Commands 190
Assignments 190
How to Smile when Disaster Strikes 190
Why Disaster May Strike 191
The DOS BACKUP Command 192
BACKUP Command Error Codes 196
What to Do After the Crash 196
Caveats to the DOS BACKUP and RESTORE Commands 198
A Typical Backup Schedule 199
Copy-Protection and the Backup Process 202
Backup--A Wearisome Exercise 203
Guided Activity: Use the Backup and Restore Process 204
Additional Notes on RESTORE 208
Guided Activity: Use the Partial Restore Process 210
Another Safety Command--ATTRIB 212
Setting the Read-only Bit 213
Setting the Archive Bit 214
The ATTRIB /S Parameter 215

Guided Activity: Mark Files Read-only with the ATTRIB Command 216
Copying Groups of Files with XCOPY 222
Guided Activity: Use the XCOPY Command 224
The REPLACE Command--A Handy Option 228
REPLACE Command Error Codes 229
Guided Activity: Use the REPLACE Command 230
Recovering the Good "Parts" of a File 236
Command Review Dictionary 238
Review Questions 239
Unit Exercises 240
Documentation Research 240

Application F 241

Part 1: Restoring an ASCII Text File 241
Part 2: Safeguarding Files with the ATTRIB Command 242

Part IV Additional DOS Commands 243

Unit 9 DOS Redirection, Pipes, and Filters 245

Objectives 245
Important Commands 245
Assignments 246
Standard PC DOS Devices 246
Redirection of Standard Input and Output 249
Guided Activity: Redirect DOS Input and Output 251
Pipes and Filters--DOS Building Blocks 259
The SORT Filter 259
The SORT Parameters 262
The MORE Filter 263
Guided Activity: Use the SORT and MORE Filters 264
The FIND Filter 271
The FIND Filter Parameters 273
Guided Activity: Use the FIND Filter 275
Command Review Dictionary 282
Review Questions 283
Unit Exercises 284
Documentation Research 284

Application G 285

Part 1: Searching for Text Phrases in a File 285
Part 2: Using Pipes, Filters, and Redirection 286

Unit 10 Screen and Keyboard Control Commands 287

Objectives 287
Important Commands 287
Assignments 288
Five Modes of Operation 288
The MODE Command--Changing Display Adapters 289
Code Page Switching 290
Side Trip to Investigate the SELECT Command 291
The MODE Command and Code Page Switching 292
The CONFIG.SYS File Revisited 295
Screen and Keyboard Control with ANSI.SYS 296
Communicating with ANSI.SYS 296
Cursor Movement and Message Displays with ANSI.SYS 299
Guided Activity: Use the PROMPT Command and ANSI.SYS to move the Cursor 301
Setting Screen Colors and Display Attributes with ANSI.SYS 306
Guided Activity: Use the PROMPT Command to Alter Screen Colors and Attributes 307
Keyboard Reassignment with ANSI.SYS 310
Guided Activity: Redefine Keyboard Keys with ANSI.SYS and the PROMPT Command 313
Command Review Dictionary 318
Review Questions 319
Documentation Research 320

Application H 321

Part 1: Creating an ANSI.SYS Screen Change Batch File 321

Unit 11 Printer Control Commands 323

Objectives 323
Important Commands 323
Assignments 324
The MODE Command--Changing Printer Output 324
The MODE Command--Switching from Parallel to Serial 325
Guided Activity: Use the PRINT Mode Option 325
Background Printing with the PRINT Command 326
Guided Activity: Create a Print QUEUE 330
Command Review Dictionary 335
Review Questions 336
Documentation Research 336

Application I 337

Part 1: Resetting Printer Output with the MODE Command 337

Unit 12 RAM Drives and Data Communications 339

Objectives 339
Important Commands 339
Assignments 340
The MODE Command--Setting the Serial Protocol 340
The DOS RAM Drive 342
Guided Activity: Use the RAM Drive 344
Command Review Dictionary 351
Review Questions 352
Documentation Research 352

Application J 353

Part 1: Creating Multiple Virtual Drives 353

Unit 13 The DEBUG Command 355

Objectives 355
Assignments 355
The DEBUG Command 356
Why Do They Teach the Hexadecimal Numbering System in College? 356
Invoking and Quitting DEBUG 359
Additional DEBUG Commands 361
The DEBUG DUMP Subcommand 362
Guided Activity: View Data with DEBUG 365
Another Method of Loading a File with DEBUG 369
Modifying Data with DEBUG 370
Programming and Modifying Data and Code with DEBUG 371
Additional DEBUG Tools 372
Review Questions 372

Unit 14 DOS Local Area Network Commands 373

Objectives 373
Important Commands 373
Assignments 374
Local Area Networks--the Wave of the Future? 374
Network Support First Offered by PC DOS 3.x 376
The SHARE Command 376
The File Control Block Command 377
Specifying the LASTDRIVE 378
Network Software Commands 381
What is Novell NetWare? 381
Using DOS Commands on a Novell NetWare System 381
Types of Novell NetWare Commands 383
Selected Novell NetWare Commands 384
Command Review Dictionary 388

Review Questions 389
Documentation Research 389

Unit 15 Multitasking, Multiprocessing, and the Future of DOS 391

Objectives 391
Assignments 391
Five User Concerns for Future DOS Directions 392
Availability of More Memory 392
Extended Memory 394
Expanded Memory 394
Multitasking 395
Multiprocessing 395
Upward Compatibility 396
User-Friendliness 396
Conclusion 397
Additional References on Developments in PC DOS and MS-DOS 397
Review Questions 398
Documentation Research 398

Appendix A Computer Terms Dictionary A-1

Appendix B FDISK and Hard Disk Formatting B-1

Appendix C Types of Disk Drives C-1

Appendix D A Concise History of MS-DOS/PC DOS D-1

Differences Between PC DOS and MS-DOS D-4
List of Internal DOS Commands D-4
List of External DOS Commands D-6

Appendix E MS-DOS/PC DOS Command Reference E-1

PUBLISHER'S NOTE

This book is part of THE MICROCOMPUTING SERIES. We are proud to announce that this unique series is now entering its fourth year, and currently includes four different types of books:

1. A core concepts book, now in its second edition, teaches basic hardware an software applications concepts. This text is titled UNDERSTANDING AND USING MICROCOMPUTERS.

2. A series on introductory level, hands-on workbooks for a wide variety of specific software packages. These provide both self-paced tutorials and complete reference guides. Each book's title begins with UNDERSTANDING AND USING . . . .

3. Several larger volumes combine DOS with three popular software packages. Two of these volumes are called UNDERSTANDING AND USING APPLICATION SOFTWARE, while the third is titled UNDERSTANDING AND USING SHAREWARE APPLICATION SOFTWARE. These versions condense components of the individual workbooks while increasing the coverage of DOS and the integration of differen application packages.

4. An advanced level of hands-on workbooks with a strong project/systems orientation. These titles all begin with DEVELOPING AND USING . . . .

Our goal has always been to provide you with maximum flexibility in meeting the changing neds of your courses through this "mix and match" approach. We remain committed to offering the widest variety of current software packages.

We now offer these books in THE MICROCOMPUTING SERIES:

Understanding and Using Microcomputers, 2nd Edition by Steven M. Zimmerman and Leo M. Conrad

OPERATING SYSTEMS

Understanding and Using MS-DOS/PC DOS: The First Steps, 2nd Edition
by Laura B. Ruff and Mary K. Weitzer

Understanding and Using MS-DOS/PC DOS 4.0
by Jonathan P. Bacon

Understanding and Using MS-DOS/PC DOS: A Complete Guide
by Cody T. Copeland and Jonathan P. Bacon

PROGRAMMING LANGUAGES

Understanding and Using Microsoft BASIC/IBM-PC BASIC
by Mary L. Howard

WORD PROCESSORS

Understanding and Using DisplayWrite 4
by Patsy H. Lund and Barbara A. Hayden

Understanding and Using Microsoft Word
by Jonathan P. Bacon

Understanding and Using MultiMate
by Mary K. Weitzer and Laura B. Ruff

Understanding and Using PC-Write
by Victor P. Maiorana

Understanding and Using WordStar 4.0
by Patsy H. Lund and Barbara A. Hayden

Understanding and Using pfs:WRITE
by Mary K. Weitzer and Laura B. Ruff

Understanding and Using WordPerfect
by Patsy H. Lund and Barbara A. Hayden

Understanding and Using WordPerfect 5.0
by Patsy H. Lund

Understanding and Using WordStar
by Steven C. Ross

SPREADSHEET PACKAGES

Understanding and Using Lotus 1-2-3
by Steven C. Ross

Understanding and Using Lotus 1-2-3 Release 2
by Steven C. Ross

Understanding and Using Lotus 1-2-3 Release 3
by Steven C. Ross

Understanding and Using SuperCalc 3
by Steven C. Ross and Judy A. Reinders

Understanding and Using SuperCalc 4
by Judy A. Reinders and Steven C. Ross

Understanding and Using VP-Planner Plus
by Steven C. Ross

DATABASE PACKAGES

Understanding and Using dBASE III (Including dBASE II)
by Steven C. Ross

Understanding and Using pfs:FILE/REPORT
by Laura B. Ruff and Mary K. Weitzer

Understanding and Using dBASE III PLUS
by Steven C. Ross

Understanding and Using dBASE IV
by Steven C. Ross

Understanding and Using R:BASE 5000
(Including R:BASE System V)
by Karen L. Watterson

INTEGRATED SOFTWARE

Understanding and Using Appleworks
(Including Appleworks 2.0)
by Frank Short

Understanding and Using Educate-Ability
by Victor P. Maiorana and Arthur A. Strunk

Understanding and Using FRAMEWORK
by Karen L. Watterson

Developing and Using Office Applications
with AppleWorks
by M. S. Varnon

Understanding and Using pfs:First Choice
by Seth A. Hock

Understanding and Using Symphony
by Enzo V. Allegretti

COMBINATION VOLUMES

Understanding and Using Application Software, V. 1:
DOS, Lotus 1-2-3 Release 2, and dBASE III PLUS
by Steven Ross, Patsy Lund and Barbara Hayden

Understanding and Using Application Software, V. 2: DOS,
WordPerfect 4.2, Lotus 1-2-3 Release 2, & dBASE III PLUS
by Steven Ross, Patsy Lund and Barbara Hayden

Understanding and Using SHAREWARE WordStar 4.0,
Application Software: DOS, PC-Write,
ExpressCalc, and PC-FILE
by Victor P. Maiorana

Understanding and Using Application Software,
Volume 3: DOS, WordPerfect 4.2, VP-Planner
Plus, and dBASE III PLUS
by Steven C. Ross and Patsy H. Lund

ADVANCED BOOKS

Developing and Using Decision Support Applications
by Steven C. Ross, Richard J. Penlesky
and Lloyd D. Doney

Developing and Using Microcomputer Business
Systems
by Kathryn W. Huff

We are delighted by the popularity of THE MICROCOMPUTING SERIES. We appreciate your support, and look forward to your suggestions and comments. Please write to us at this address:

West Publishing Company,
College Division
50 West Kellogg Blvd.,
P.O. Box 64526, St. Paul, MN 55102

ABOUT THE AUTHORS

Cody T. Copeland holds a BA from the University of Arizona at Tucson and an MA from the University of Wyoming at Laramie. He has been involved in technical training for the past 14 years in the fields of aviation and microcomputers.

Copeland served as director of continuing education at Johnson County Community College (JCCC) at Overland Park, Kansas from 1982 until 1988. During that time he founded and developed the JCCC Microcomputer Training Center. The center currently trains over four thousand business and industry students each year. Copeland has written training materials for courses on Symphony, Introduction to Personal Computers, Lotus 1-2-3, the Smart Software System, and WordPerfect.

Copeland, is currently a doctoral student at the University of Kansas in Educational Policy and Administration. In addition, he serves on the JCCC faculty and teaches in the Computer Information Systems division. He also has provided consulting for community agencies, business, and industry on DOS, desktop publishing, and WordPerfect.

Jonathan Bacon holds a BA in English literature and an MA in student personnel work, both from Michigan State University at East Lansing. He has also completed a wide range of studies in the data processing area. His experience includes curriculum design for several Computer Application Program credit courses offered at JCCC.

Bacon was instrumental in developing the curriculum for the JCCC Microcomputer Training Center. He has written training materials for courses on Lotus 1-2-3, PC DOS, Multimate, MultiPlan, PFS:File, PFS:Write, the Smart Software System, SmartWareII, and WordStar.

In addition to the current text, Bacon has two other West Publishing textbooks to his credit: *Understanding and Using MS-DOS/PC DOS 4.0* and *Understanding and Using Microsoft Word*. He is also a regular columnist for *SMARTIMES*, a personal computing journal, and is the author of a trade paperback on SmartWareII.

Bacon is currently manager of user information systems at JCCC. He formerly served as director of student development and counseling at the college. His current responsibilities include micro and mainframe support, training, consulting, and programming for JCCC programs and staff. He also has provided consulting for community agencies, business, and industry.

PREFACE

WHO SHOULD USE THIS BOOK?

Welcome to *Understanding and Using MS-DOS/PC DOS: A Complete Guide*. This book is designed to meet the needs of both students and instructors, both novice and experienced users, who seek to learn more about DOS. Whether you are a student, instructor, corporate manager, secretary, or home user, this book is intended to familiarize you with the mysteries of the IBM personal computer operating system (PC DOS and MS-DOS).

Our aim is to introduce you to new concepts regarding DOS, one at a time. After some explanatory text, we'll follow up with step-by-step Guided Activities to make the new concepts and skills clear. We believe in hands-on learning--with computers there is no other way.

CONTENT HIGHLIGHTS

Understanding and Using MS-DOS/PC DOS: A Complete Guide is divided into four parts. Part I introduces you to operating systems and explains what they are. Part II explores the fundamental DOS commands including commands necessary to format disks, manage files, and organize files on a hard disk. Part III includes an exploration of intermediate DOS commands used to edit files, automate computing, and safely backup and restore your data. Part IV introduces you to several additional DOS commands and concepts such as pipes, filters, and redirection. Additionally, Part IV covers printer and screen control commands, RAM drives, the DEBUG command, and provides an overview of local area network commands. The final unit in Part IV, discusses multitasking, multiprocessing, expanded and extended memory, and user needs for the future of DOS.

You may be tempted to overlook Part I because it is straight reading. Please resist that temptation unless you have had previous exposure to the basics of DOS. Keep in mind that the terms defined and the concepts explained in that part of the book are essential to understanding the subsequent Guided Activities.

With the exception of Part I, this book is meant to be read while seated in front of your computer, so you can complete the step-by-step Guided Activities. Each unit includes several important segments:

Objectives
Important Commands
Assignments
Command Review Dictionary
Unit Exercises
Review Questions
Documentation Research

Additionally, most units include one or more Guided Activities preceded by explanatory text. Unit 1 discusses the hardware and software needed to complete the Guided Activities in the text. In summary, the student or reader should have DOS 3.x installed on an IBM PC, XT, AT, PS/2, or compatible with at least a single floppy drive and a hard-disk system. Blank diskettes appropriate for the system you are using will be required.

The Documentation Research activities require access to the *IBM Disk Operating System Version 3.30 Reference* manual or *Microsoft MS-DOS User's Guide and User's Reference*. These manuals are part of the documentation provided with PC DOS or MS-DOS 3.3.

If using this text in a classroom setting, your instructor will indicate which assignments, review questions, and documentation research will be assigned for the units covered.

APPROACH TO THE TEXT

This text has been constructed to allow the instructor and student to select the units and assignments most appropriate for their needs. Each unit includes numerous assignments, review questions and documentation research. The text is written to build from scratch with each subsequent unit building on concepts presented in previous units. However, a pick-and-choose approach may be just as appropriate for students with previous computing experience.

COMPUTER TERMS DICTIONARY

Appendix A is a comprehensive dictionary of operating system terms and concepts. Without turning the entire text into a dictionary, it is not always possible to define terms as they are introduced--though we have tried to do so. We have used

boldface to alert the reader to the first occurrence of terms appearing in the Computer Terms Dictionary (Appendix A). Where appropriate, dictionary terms to be reviewed are given in the Assignments section of each unit.

VERSIONS OF DOS

Screens and examples used throughout this text are based on PC DOS version 3.3, though the screens should be almost identical to those based on all versions of DOS 3.x. Software designers, developers, and users typically use the "x" to designate any version in the series. For instance, "3.x" refers to any version of DOS that composes the third major revision (which includes only 3.10, 3.20, and 3.30). Many of the Guided Activities within the units can be accomplished using any version of DOS 3.x, though DOS 3.3 is the preferred version. It contains some enhancements not found in earlier versions.

CHANGES FROM THE 1987 EDITION

The state of computing has changed since the 1987 edition of this text was written. At that time DOS versions 3.2 and 3.3 had not been released, most college training labs included dual floppy-based systems, and the 3½-in. floppy disk for PC DOS and MS-DOS machines had not been introduced.

Today, the 3½-in. disk is almost as common as its 5¼-in ancestor, most training facilities include computers with hard-disk subsystems, and DOS 3.3 has added several important new commands and enhancements.

Unlike its predecessor, this text assumes the availability and use of a hard disk. It also provides information on the entire range of floppy disks now available. Screen images have been updated to reflect DOS version 3.3. Units dealing with the hard disk and its organization have been expanded and enhanced. This text also includes coverage of several new commands including XCOPY, REPLACE, and RECOVER. A new unit on DEBUG has been added, and the unit on network commands has been expanded to include examples from the Novell NetWare software system. Coverage of the ANSI.SYS device driver has also been added.

Much of the information previously covered in Part I has been made optional and moved to an appendix. This includes information on types of disk drives, a history of the MS-DOS and PC DOS operating systems, and a command reference of all common DOS command forms.

As with the earlier text, the emphasis is still on hands-on Guided Activities. With any computing topic, you learn by doing.

ACKNOWLEDGMENTS

Many individuals have contributed to this text. Among those assisting was an outstanding group of reviewers who carefully scrutinized several drafts of the manuscript.

REVIEWERS OF THE FIRST EDITION BEFORE PUBLICATION

Terry S. Dorsett, Central Arizona College; Jack T. Hogue, The University of North Carolina at Charlotte; John Morack, Waukesha County Technical Institute; George Upchurch, Carson-Newman College; Jack van Luik, Mt. Hood Community College.

REVIEWERS OF THE FIRST EDITION AFTER PUBLICATION

Beryl Ackerman, Queens College; James Bode, Manatee Community College; John Brown, Western State College of Colorado; Robert Brown, University of Georgia; James Heffernan, University of the Pacific; Marlyce Johnson, Milwaukee Area Technical College; Jerry Joyce, Keene State College; Judy Mazzeo, Lakeland Community College; Margaret Zinky, Phoenix College.

REVIEWERS OF THE SECOND EDITION BEFORE PUBLICATION

John Brown, Western State College of Colorado; Gary Gora, Northern Alberta Institute of Technology; James Heffernan, University of the Pacific; Judy Mazzeo, Lakeland Community College; Thomas Wallin, Southern Business School.

ADDITIONAL THANKS

Additionally, the authors would like to express their appreciation for the support of the following West Publishing Company staff.

Rick Leyh and Rich Wohl, Editors; Susanna C. Smart, Developmental Editor; Connie L. Austin, Ancillary Publications Assistant; and Sherry Goldbecker, Copy Editor.

Copy T. Copeland would like to thank Susan, Tyler, Matthew, and Jeremy Copeland; "who have allowed a hobby to consume me. Also special thanks to my Mother, Lucille F. Copeland."

Jonathan Bacon would like to thank "the three young women who provide inspiration in my life; Joan, Jennifer, and Jodi Bacon. Also a special thanks to my Father, Paul J. Bacon for his continual support."

PART I

OVERVIEW OF MS-DOS AND PC DOS

This is the first of four parts in this text. In Part I, we offer a general overview of IBM PC DOS and its sister operating system, MS-DOS. We discuss the basic function of a microcomputer operating system, computer language syntax, and internal versus external DOS commands. This first part concludes with a discussion of basic concepts of DOS including filenames, extensions, wildcard characters, and drive designations. Once this background information is covered, we can proceed to discuss basic DOS commands in Part II.

UNIT 1

AN INTRODUCTION TO OPERATING SYSTEMS

OBJECTIVES

After completing this unit, you will be able to

1. explain the function of a microcomputer disk operating system;
2. explain the process followed when a computer is booted;
3. explain why computer language syntax is important;
4. define the difference between internal and external DOS commands;
5. list the guidelines for creating filenames under DOS;
6. define the different parts of a filename;
7. recognize acceptable DOS filenames;
8. indicate the function of the DOS prompt;
9. list the keystrokes required to change the default drive;
10. explain how DOS commands are issued.

ASSIGNMENTS

The assignments to be completed for this unit are

_____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): archive, BASIC (Beginner's All-purpose Symbolic Instruction Code), basic input/output system (BIOS), boot, clones, compatibles, complete filename, CPU, default drive, delimiter, DOS (Disk Operating System), DOS prompt, drive designation, extension, external DOS commands, filename, file specification, firmware, input, internal DOS commands, kilobytes (K), megahertz (Mhz), MS-DOS, OS (operating system), Operating System/2 (OS/2), output, parameters, path, PC (personal computer), PC DOS, peripherals, Personal System/2 (PS/2), POST (power-on self-test), RAM (random access memory), reserved characters, ROM (read-only memory), source disk, switch, syntax, system prompt, target disk, and wildcard characters.

_____Review Questions.

_____Documentation Research.

Place a check in front of those assigned by your instructor.

AN INTRODUCTION TO OPERATING SYSTEMS

Computer users are usually familiar with the types of business software on the market; that is, word processors, spreadsheets, business graphics, and data base management software. Additional application programs exist to facilitate telecommunications, time management, accounting, payroll management, inventory, scheduling, desktop publishing, and a host of other functions.

Few people get excited about one category of software, yet without it, application software is useless. Without this forgotten software the **central processing unit (CPU)**, the heart of the computer system, cannot communicate with attached **peripherals** (such as printers and modems); **input** cannot be accepted from the keyboard, and **output** cannot be displayed on the monitor screen.

New microcomputer owners have been known to cart home a system unit, monitor, printer, cables, and tons of software, and yet neglect to purchase this oft-forgotten software.

On the surface, microcomputers appear to be miraculous little machines with secretive insides. It may seem as if putting a floppy disk in the drive and hitting the appropriate key is the totality of this new technology. There is neither magic nor mystery in the system unit. Microcomputers need instructions in order to function. Meticulous, step-by-step instructions must tell the micro how to behave like a computer.

In the world of mainframe computers, these instructions are referred to as the **OS** (operating system). The OS must begin operation the second the computer system is powered up and handle additional instructions as required. The computer must know how to respond to keyboard input, read directions from memory or a storage device (like a diskette or hard disk), and wait for further instructions.

From a mainframe point of view, a subset of the operating system is the **DOS** (disk operating system). Mainframe DOS handles only the basic input and output operations between the CPU and the disk drives. But in the world of microcomputers, the operating system has become synonymous with DOS, which rhymes with "boss." When referring to DOS, most microcomputer users are talking about the entire operating system.

The IBM **PC** (personal computer) family of microcomputers and **compatibles** use what is referred to as IBM **PC DOS** (Personal Computer DOS) or **MS-DOS** (Microsoft DOS). Microsoft is the firm that initially developed PC DOS for IBM computers and MS-DOS for IBM compatibles.

#3

Word processors, data managers, accounting programs, and all other software are written to run with a specific operating system. Operating systems in turn are written to take advantage of a specific microprocessor. The following machines use the microprocessors listed:

Microcomputer	Microprocessor chip
Apple	MOS Technology 6502 and 65C02
Commodore Amiga	Motorola 68000
IBM PC, XT, AT, PS/2	Intel 8088, 80286, and 80386

A software program written for the Apple or other non-IBM compatible machines will not run on the IBM PC unless the program is rewritten to use the IBM operating system and hardware. The reverse is also true. PC DOS and MS-DOS were designed to use the features of the 8088 microprocessor and its descendants.

#4

Compatibles exist because their manufacturers use the same basic operating system, or one which closely duplicates the operating system's set of instructions. Part of DOS, called the **basic input/output system** (BIOS), is contained in **ROM** (read-only memory). ROM microchips contain circuitry patterns with embedded instructions. Each time electric current moves through the circuitry, the same functions are performed. Some of these ROM chips include the instructions that form the BIOS, or the interface between the OS and the specific hardware being used. This portion of the OS is so automatic that most users rarely concern themselves with it.

BOOTING YOUR SYSTEM AND THE POWER-ON SELF-TEST

An example of a ROM-based portion of DOS is the **POST** (power-on self-test). When an IBM PC is initially started (**booted**), it conducts a power-on self-test that takes 3 to 90 seconds [depending on how much **RAM** (random access memory) is installed in the computer, the speed of the microprocessor, and the kind of hardware components installed].

NOTE: The process of starting up or turning on your computer is referred to as *booting your system*. The term comes from the concept of "pulling oneself up by the bootstraps." The expression means the person is a self-starter and tackles tasks without the assistance of anyone or anything.

Microcomputers entered the computing arena as free-standing, self-starting terminals. They booted themselves independent of any mainframe computer. Prior to 1981, you worked on a terminal connected to a mainframe (hidden away in some large room). If the system was "down" or out of commission, you were stranded with no computing power. In that light, PCs are like people "pulling themselves up by their bootstraps"--they do not depend on a larger computer system to boot. Thus, the term booting your system evolved.

The POST checks memory and tries to assure that the system is operating properly before proceeding. Next the POST checks the hardware configuration to determine, for example, if a keyboard and printer are attached. With an IBM microcomputer, if everything is functional, the system generates a single beep. A second beep means a problem was encountered, such as a bad memory chip, an unattached keyboard, or some other problem. PS/2s will display an error message in addition to the double beep. IBM compatibles respond differently. They may not beep, but only display an error message on the monitor screen.

The instructions for POST are included in ROM. Time after time, the computer will follow these instructions when it is booted. The POST is part of the operating system.

After the POST routine is completed, DOS instructs the computer to check disk drive A: for instructions. You will note that the in-use light on drive A: lights up at this point. If instructions are found on a disk, the computer reads them. If no disk is in the drive (and the system includes a hard disk), then the BIOS checks drive C: (the hard disk) for instructions. On the IBM PC, XT, and AT computers, if computer-readable instructions are still not found, the BIOS takes over and reads a version of IBM Personal Computer **BASIC** (Beginner's All-purpose Symbolic Instruction Code) into memory. This version of BASIC is stored in ROM.

NOTE: With the introduction of the IBM PS/2 models, BASIC is no longer included in ROM. Instead, the PS/2s display a screen message prompting you to insert a diskette. When booting PC compatibles (also called **clones**), other responses may be encountered. The most common is an error message indicating that no disk is in the disk drive. Clone manufacturers do not include ROM BASIC because it is proprietary **firmware** (software on a microchip). Further, a version of BASIC is packaged with PC DOS and MS-DOS.

GETTING STARTED BY LOADING DOS

When the computer goes to drive A: and then subsequently to drive C:, it is looking for additional instructions or additional specialized portions of the software called DOS.

Loading DOS means that some of the DOS commands (referred to as **internal DOS commands**) are read from disk into the computer's working memory. You must load DOS before you can use any application software or any of the system's utilities. If you load DOS and take no further action, the computer is said to be *at the DOS level*, and the **DOS prompt** will appear on your monitor screen. In other words, you can execute DOS commands, but no application program is yet loaded.

#5

Some application software will require you to use the DOS disk to load the operating system before loading the application program. Other software programs will provide instructions on how to install DOS on your software program disk. By installing DOS on your program disk, you can make the program self-loading and give yourself access to the operating system without having to switch disks. However, with the increased use of hard-disk systems, it is even more practical to load DOS and all application software directly on the hard disk. In such a case, you actually boot from the hard disk and do not need to swap diskettes to run various programs. Data is also stored on the hard disk. The first Guided Activity in this text will walk you through the process of booting your computer and arriving at the DOS prompt.

TYPES OF PERSONAL COMPUTERS

The original IBM PC was based on the 8088 microprocessor, as was the original IBM XT. Subsequently, IBM released the Enhanced XT and the AT, both using the 80286 microprocessor.

Figure 1-1 shows a typical IBM PC setup including a system unit with two disk drives (side by side), monitor, keyboard, and attached printer. By the advent of the PC AT, the floppy disk drives in two-disk-drive systems were typically stacked rather than side by side. Stacked disk drives are referred to as *halfheight drives*. They are half the height of the original PC drives, and two of them fit nicely in the slot left by a single PC full-height drive.

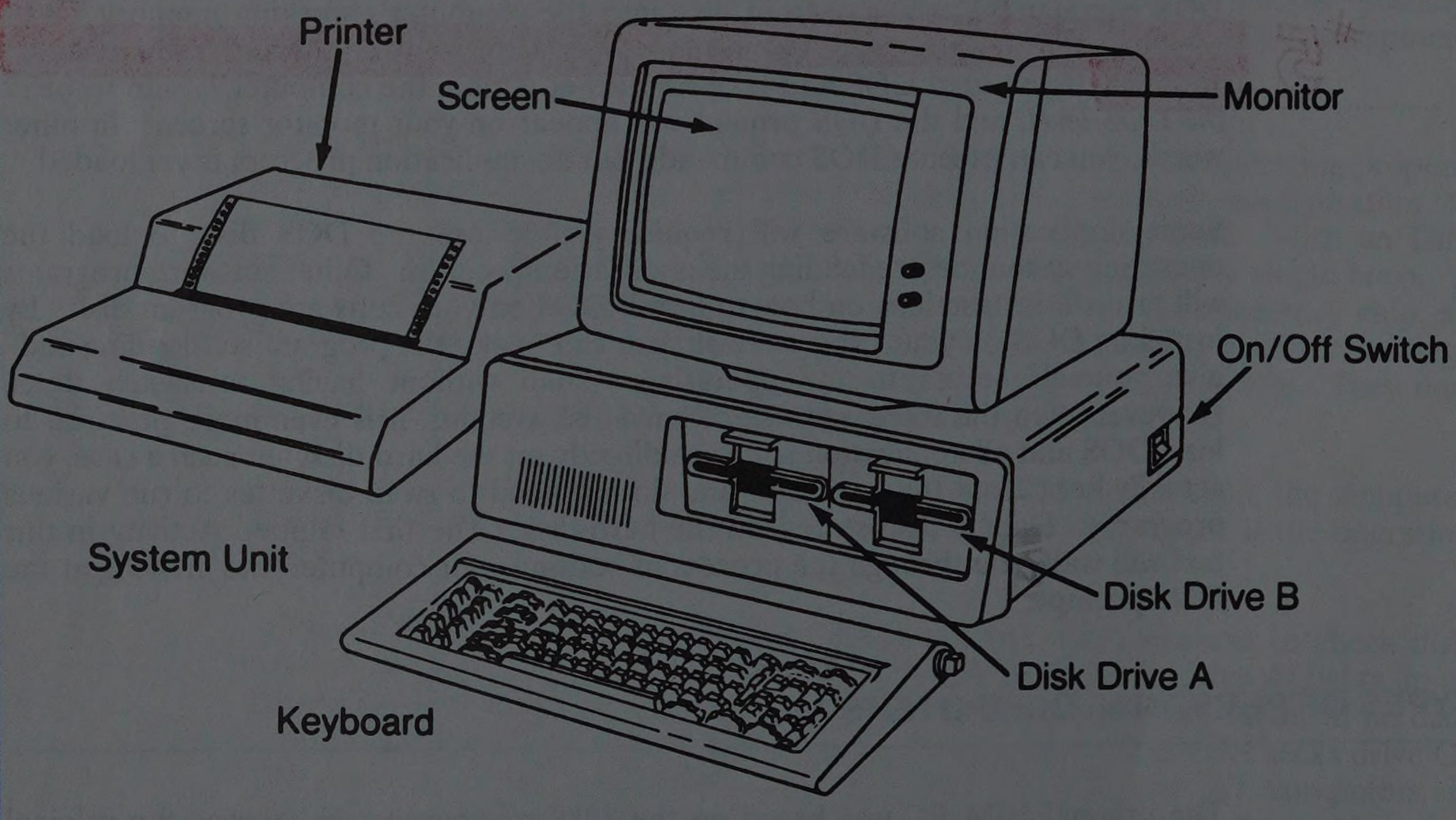


Fig. 1-1. Typical IBM personal computer configuration.

In April 1987, IBM decided to abandon the old PC architecture and instead proceed with manufacturing and marketing the new **Personal System/2** (PS/2) family of computers. The earlier PC, XT, and AT are now referred to as Family I computers, and the new PS/2s are part of the Family II series of micros. Figure 1-2 illustrates a typical PS/2 system with a single 3½-in. disk drive and an internal hard disk.

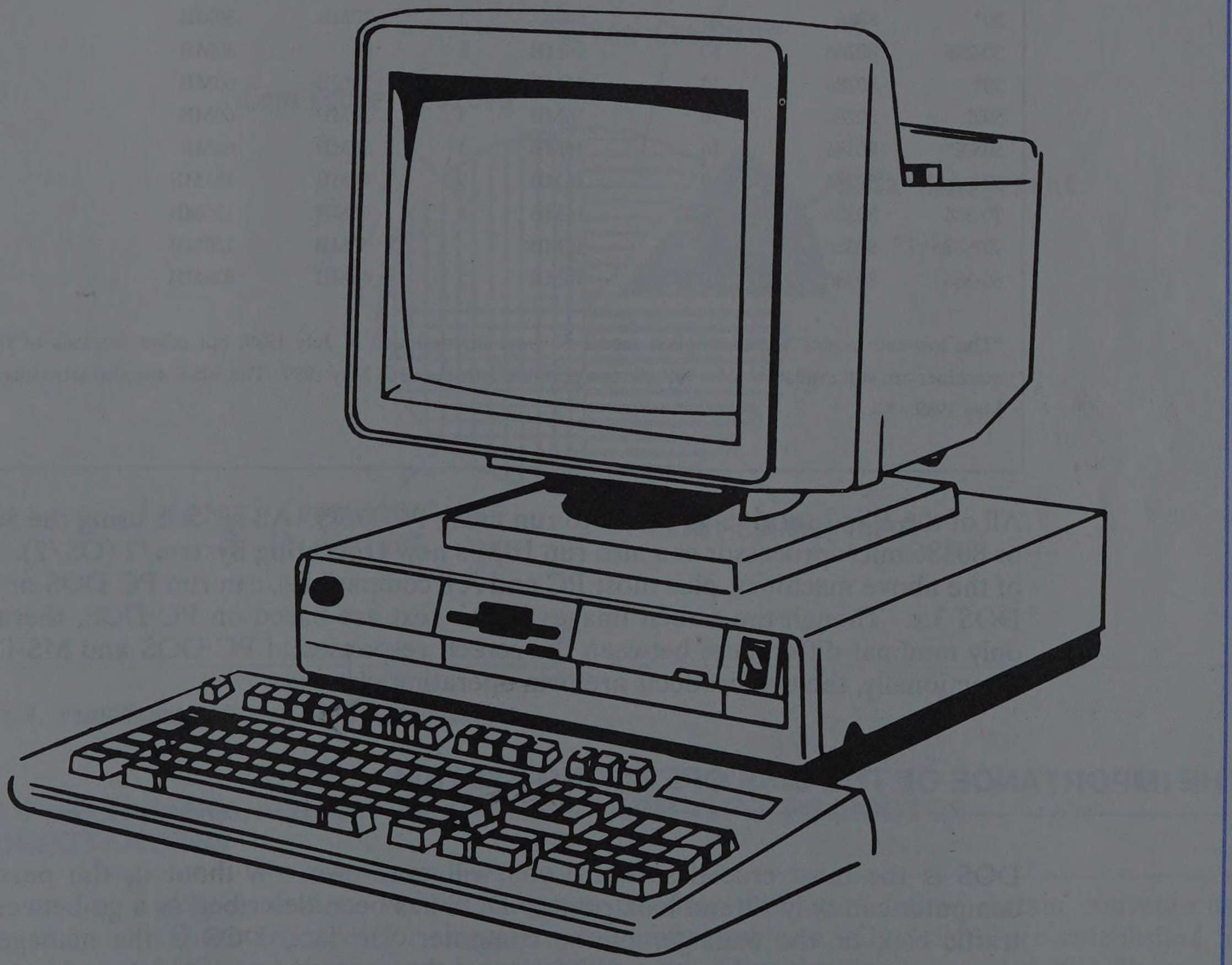


Fig. 1-2. Typical IBM PS/2 system setup.

IBM continues to release new models to fill in the gaps in the Family II series. At this time, the machines listed below are available. In this listing, you will find information on the microprocessor (Processor) used by each system, the speed in **megahertz** (Mhz), the maximum system memory (RAM), the number of slots, the standard hard-disk (Strd HD) configuration, and the maximum hard-disk (Max HD) capacity. These figures are supplied by IBM and may change as new systems are released and old ones are enhanced.

Model	Processor	Mhz	RAM	Slots	Strd HD	Max HD
25	8086	8	640K	2	0	20MB
30*	8086	8	640K	3	20MB	30MB
30-286	80286	10	16MB	3	0	30MB
50*	80286	10	16MB	3	20MB	60MB
50Z	80286	10	16MB	3	30MB	60MB
50SX*	80386	16	16MB	3	30MB	60MB
60	80286	10	16MB	7	44MB	185MB
70-386	80386	16-25	16MB	3	60MB	120MB
70P-386*	80386	20	16MB		60MB	120MB
80-386	80386	16-20	16MB	7	44MB	628MB

*The low-end model 30 and original model 50 were discontinued in July 1989, but other versions of these machines are still available. The 70P-386 is a portable introduced in May 1989. The 50SX was also introduced in May 1989.

All of the PS/2 models are built to run using PC DOS. All models using the 80286 or 80386 microprocessor can also run IBM's new **Operating System/2** (OS/2). Any of the above machines, plus most PC and AT compatibles, can run PC DOS or MS-DOS 3.x. Though the screen images in this text are based on PC DOS, there are only minimal differences between the screen responses of PC DOS and MS-DOS. Functionally, the two products are twin operating systems.

THE IMPORTANCE OF THE DISK OPERATING SYSTEM

DOS is the most crucial software you will ever own. Without it, the personal computer can only sit and look pretty. DOS has been described as a go-between, a traffic cop, or the manager of the computer. In fact, DOS is the manager of communications (or data transfer) between the computer and its internal/external parts such as memory, printers, disk drives, and video display. Figure 1-3 shows some of the functions performed by the operating system.

The operating system handles basic instructions such as telling the computer how to retrieve, store, or delete data from the disk. These instructions are written in machine language, yet you can perform similar operations by using DOS commands such as COPY, DISKCOPY, and ERASE (these commands are introduced in later

units). Software developers also execute these commands from within their products to copy, delete, or save data.

Even if you intend to only use prepackaged software, a basic understanding of the DOS commands is essential. Without understanding DOS, you cannot do even simple maintenance activities such as making duplicates of data or application program diskettes.

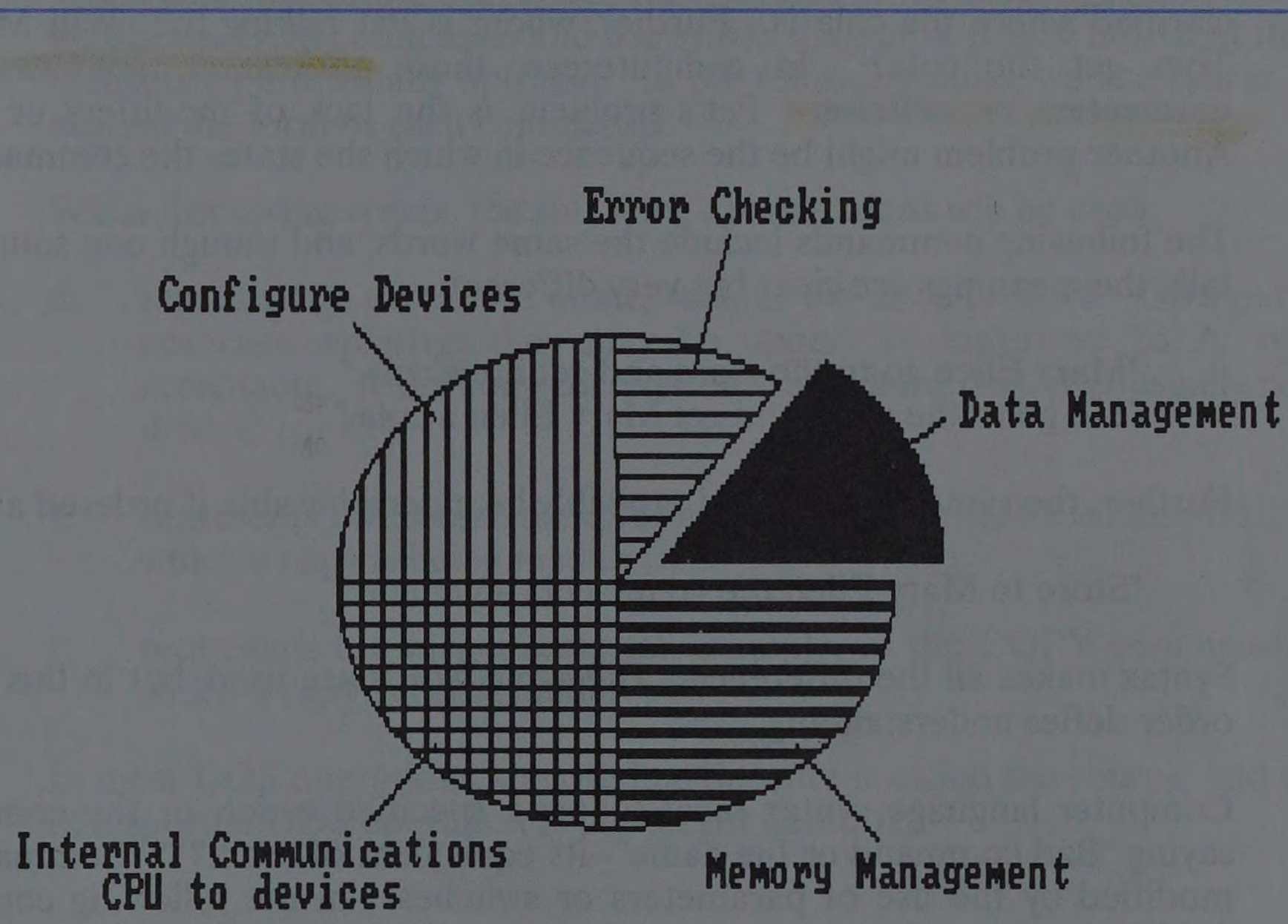


Fig. 1-3. Functions of the operating system.

COMPUTER LANGUAGE SYNTAX: THE ORDER THAT PROMOTES UNDERSTANDING

Any spoken or written language requires that users follow certain conventions. There are rules for singular versus plural subjects. Guidelines are established to define what is present tense or past tense. Even the **syntax** or order of the language is important.

Syntax is simply the way in which words, phrases, and ideas are put together, either verbally or on paper. We use language to communicate with computers; and because these machines are limited in understanding, the syntax with computers is more exacting. They cannot guess our intentions based on body language, voice inflection, or familiarity.

An example of using syntax in the English language might clarify its use with computers. Mary Ellen, Pat, and Tom are watching television when Pat becomes overwhelmed by a thirst for a cola. She knows there are no soft drinks in the refrigerator. Pat says "Get," but the obvious question is "Get what?" To clarify the command, Pat says "Get a cola." Further clarification is still needed. Is she offering Mary Ellen and Tom a cola, and if so, where are the colas?

The command must be expanded to include "Get me a cola," and yet she still has not clarified where the cola is. Further, whom is Pat talking to? Will Mary Ellen or Tom get the cola? In computerese, those additional modifiers are called **parameters** or **switches**. Pat's problem is the lack of modifiers or parameters. Another problem might be the sequence in which she states the command.

The following commands include the same words, and though one sounds like baby talk, the meanings are clear but very different:

"Mary Ellen go to the store and get me a cola."
"Me go to the store and get Mary Ellen a cola."

Further, the same words would probably be indecipherable if ordered as

"Store to Mary Ellen the go me cola a get and."

Syntax makes all the difference. The same words are used, but in this phrase their order defies understanding.

Computer language syntax must follow a specified order, or the computer is left saying "Bad command or file name"--its equivalent of "Huh??" Commands must be modified by the use of parameters or switches. In the following command form (which is also referred to as a command line), everything after the command "COPY" is a parameter or switch:

```
COPY a:FILE-A b:FILE-A
```

It says to the computer, copy the file named FILE-A from the disk in drive A: to the disk in drive B:. It might be helpful to note that with DOS if two **filenames** are listed on a command line, the first is always the source file and the second is the target. This is a PC DOS/MS-DOS syntax convention. The first file is the FROM file and the second is the TO file. So, in the above example, the copy is FROM a:FILE-A TO b:FILE-A. If a file is being renamed, the REN command could be used. Changing the filename from HELP to ASSIST would require the following command line:

```
REN a:HELP ASSIST
```

NOTE FOR FORMER CP/M USERS: For readers who may be familiar with the CP/M operating system, the syntax in PC DOS and MS-DOS is the reverse of what you might expect. CP/M used the syntax standard of COPY TO FROM rather than the DOS convention of COPY FROM TO. Just be aware of the difference.

When communicating with DOS, certain rules of the language must be followed. As we discuss DOS commands and use syntax diagrams, please note that items listed in brackets [ ] are usually optional. In the following units, we use syntax diagrams to explain the form of each command.

When discussing syntax, the following abbreviations will be used:

d: represents a disk drive name, such as the a: or b: drive. DOS parameters are not **case-sensitive**; they may be upper- or lowercase so A: or B: is also acceptable. If the computer system has a hard disk, it is usually referred to as drive c: (or C:).

s: represents the **source disk**; for example, in the COPY command, the disk from which a copy is being made.

t: represents the **target disk**; for example, in the COPY command, the disk to which a copy is being made.

In most DOS operations, the original diskette is called the source, and the recipient of a specific DOS operation is the target diskette.

ISSUING DOS COMMANDS

Several guidelines on issuing DOS commands may be helpful as you proceed through the following units.

1. A DOS command must be spelled correctly using the syntax described throughout this text, including any spaces within the command line. A space will follow each command or filename just as words in English are separated by spaces.

2. Commands may be typed in either uppercase or lowercase. For clarity, DOS commands are printed in uppercase in this text, but they are not case-sensitive. Capitalization is not required. Further, to clarify spacing, text to be entered looks more like typewritten characters than typeset characters in this text (such as `COPY A:*.* B:`).

3. After typing a DOS command line, you must press <ENTER>. This tells the computer to begin executing the command.

4. If the command you type contains a typographical error, a missing space, an extra space, or other error (such as indicating the wrong drive), the error message "Bad command or file name" will appear when you press <ENTER>. If this prompt does appear, simply retype the command correctly and try again.

INTERNAL VERSUS EXTERNAL DOS COMMANDS

Perhaps the most difficult concept for the novice PC user is the differences between **internal DOS commands** and **external DOS commands.** Appendix D includes a list of internal and external DOS commands for each version of DOS.

When we discuss filename extensions, you will find that files with a .COM or .EXE **extension** on the DOS disk are external commands. In this usage, external simply means that the command must be read from disk in order to be executed. It is external to working memory (RAM) until the file is read into memory.

Internal DOS commands are loaded when the system is booted and are always resident in working memory. By working memory, we mean random-access memory (RAM). The original IBM PC came with 64K of memory. The more recent Revision "B" PC includes a minimum of 256 **kilobytes** (K) of memory and can be upgraded to 640K. 640K is the maximum memory addressable at one time by the 8086 or 8088 microprocessor and versions of DOS including 3.x. A portion of that memory is consumed by loading DOS. The COMMAND.COM file includes instructions for the DOS command processor (i.e., that "thing" that processes commands). With PC DOS, COMMAND.COM automatically loads IBMBIO.COM and IBMDOS.COM, two hidden files which contain additional portions of DOS (these two files are called IO.SYS and MSDOS.SYS in versions of MS-DOS). Once they are loaded, the internal DOS commands are accessible from the keyboard.

NOTE: Hidden files are files that will not show in a directory listing when you invoke the DIR command. DIR is one of the first commands you learn in Unit 3.

The following analogy (shared by Professor Kevin R. Parker of New Mexico Junior College) might help clarify internal versus external DOS commands. Internal DOS commands are like words that you know. When you hear a familiar word, you know its meaning because it is stored in your mind. External DOS commands are like unfamiliar words. To understand (process) the word, you must look it up in a dictionary, just as DOS must "look up" an external command on the DOS diskette or in the DOS directory on a hard disk.

AN IMPORTANT NOTE: Both internal and external DOS commands are available only when the DOS prompt (A> or B> or C>) is displayed on the screen. That prompt literally means "ready for your command."

FILENAMES, EXTENSIONS, AND DRIVE DESIGNATORS

DOS stores all information it is given (data or programs) in files. In order to save, modify, or retrieve the file, DOS must know its name. Guidelines for filenames must be understood before proceeding to discuss specific DOS commands. Even beginning computer users must use filenames--they are unavoidable. Every time data are saved to disk or an application program (like a word processor) is loaded, a filename must be used. The **complete filename** is referred to as a **file specification** and follows this form:

[d:][path]filename[.ext]

The file specification is composed of four parts:

1. the drive designation (d:)
2. the pathname (path)
3. the filename
4. the extension (.ext).

THE FILENAME: NAMING CONVENTIONS

DOS 2.1 and later versions use the following guidelines for the filename. Filenames are used with DOS to identify specific data, text, or application files. A filename must be no longer than eight characters. The following groups of characters may be used in a filename or extension:

uppercase alphabetic characters	A-Z
lowercase alphabetic characters	a-z
numeric characters	0-9
special characters	! @ # $ % & ( ) - _ { } ' ‘

It is important to know that filenames cannot include spaces or blanks. Spaces are used as **delimiters** between a DOS command and any parameters that might be needed (such as one or more filenames). Unless a DOS command is entered without any parameters, it is always followed by a space.

NOTE: You will encounter two exceptions to this rule as you learn DOS. First, a command without any parameters (such as DIR) does not have to be followed by a space. Second, any command followed by a back-slash (\) or slash (/) character does not require a space. Examples would include commands like DIR/P and CD\DOS. Don't worry about the meaning of these commands until they are introduced in later units.

Reserved characters (that is, characters that are illegal in a filename because they are reserved for other purposes in DOS) include

asterisk	*
back-slash	\
blanks or spaces	
colon	:
comma	,
concatenation character	|
double quote	"
equal sign	=
greater-than sign	>
less-than sign	<
period	.
plus sign	+
question mark	?
semicolon	;
slash	/
square brackets	[ ]

Fig. 1-4. DOS reserved characters.

Following the above guidelines, acceptable filenames could include

&G#$DX98
HUH!
PICKLES

Filenames should describe the contents of the file they identify. Thus, PICKLES would be an acceptable filename for a letter of complaint regarding an off-color jar of pickles purchased at a local supermarket. The first two filenames are not very descriptive of the file's contents, even though they are acceptable filenames. Special characters are often avoided in the filename, unless they contribute to the descriptive nature of the filename.

NOTE: There are also a few **reserved words** in DOS. You may not call a file by these names: AUX, CLOCK$, COM, CON, LPT, LST, NUL, or PRN.

THE FILENAME EXTENSION

#80

The extension is a one-, two-, or three-character add-on to the filename. It is not required, but is often helpful. The extension always follows a delimiter (a period) and is tacked onto the end of the filename. Acceptable or legal filenames with extensions would include

LETTER.DOC
COST.WS
ORDER.BAS

Filenames are not required to be eight characters long, nor are extensions required to be three characters long. Those are maximum limits. Since filenames cannot exceed eight characters, you must often truncate words to fit the character limitation. A memo detailing the company's equipment replacement schedule could be saved under the filename EQUIPMNT, EQREPLAC, or EQSCHED. In each case, the topic of the memo is truncated and reflected in the filename. If you attempt to save a file with a filename longer than 8 characters, DOS ignores the additional characters and truncates the filename at 8 characters. Many application programs prevent you from even attempting to save a filename over 8 characters in length.

Some software programs automatically attach extensions to the files created by the user. For example, word processors often append a .DOC extension to documents created. The user does not even see the extension when files are listed from within these word processors. When an application program automatically backs up files, a .BAK extension is often added to revised text files.

Various spreadsheet programs automatically add file extensions to the worksheets created, including .WKS or .WS. Another common spreadsheet extension is .DIF, a standard devised by VisiCalc that enables spreadsheet files to be transferred from one spreadsheet program to another. When files are printed to disk rather than to a printer, a typical extension used is .PRN.

Data base managers often assign either .DB or .DBF for data base files. Other associated files may use such extensions as .DFR for a report definition, .DFQ for a query definition, and .PIX for a physical index file.

In each case outlined above, the extension is automatically added by the software to keep files straight. These extensions are seldom seen within the software; only the first eight characters (i.e., characters prior to the delimiter) are displayed. Because the extensions are automatically added by the software, you can assign a worksheet the filename BUDGET and still generate a graph and perhaps a text file called BUDGET. The computer knows which is which because of the automatically generated different extensions. Respectively, the names might be BUDGET.WKS, BUDGET.PIC, and BUDGET.TXT.

DOS FILENAME EXTENSIONS

DOS has its own set of extensions, many of which are used by specific software packages, as mentioned above. The most common DOSassigned file extensions are

.BAK for a DOS-created backup file
.BAS for a BASIC language file
.BAT for a batch processing file
.BTX for a DOS-created backup text file
.COM for an external DOS command or program file
.EXE for an executable program or command file
.HEX for a file used with the DEBUG command
.LIB for library files used with the LINK command
.MAP for a file used with the LINK command
.OBJ for an object code file created by a compiler
.REL for an object code file created by an assembler
.TMP for a temporary file created by DOS

As a side note, the .COM files on the DOS diskette are really special communication files. They correspond to the standard DOS external commands already discussed. Each .COM file contains raw machine language code (instructions).

Whenever a DOS external command is entered (e.g., CHKDSK, DISKCOPY, or DISKCOMP), DOS immediately checks to see if there is a .COM file by that filename on the disk in the default or specified drive. If such a file is located, the machine language instructions contained in the file are loaded into memory, and control of the system is passed to those instructions.

IMPORTANT NOTE: If no .COM file is found, then DOS checks first for an .EXE file and then for a .BAT file by the name entered. To execute files with these three types of extensions (.BAT, .COM, and .EXE), the extension does not need to be entered, only the filename.

THE DRIVE DESIGNATION

In the earlier discussion of syntax, it was pointed out that Mary Ellen and Tom needed to be told where to go to obtain a cola for Pat.

Similarly, DOS must know where commands are located, where files are stored, and where specified actions are required. A personal computer typically has one or two disk drives. Either the lefthand drive or the top drive (if the drives are stacked) is drive A. The formal name is A: (the colon is a required part of the name, though in speaking we refer simply to drive A).

If a second disk drive is included with the computer, it is referred to as drive B:. On an enhanced PC, IBM XT, AT, or PS/2, the first hard-disk drive is referred to as

drive c: or C:. Depending on the configuration of your PC system, you may have additional drives labeled D: through Z:.

Referring to the cola example, we might say "Get a cola," but "Go to the store and get a cola" is more precise. The computer needs similar clarification. If a program is to be loaded, is it on the disk in drive A: or B: or C: or D:? The **drive designation** accomplishes this task. It communicates to the computer where commands to be executed are located.

THE DEFAULT DRIVE OR LOGGED-ON DRIVE

DOS makes life easy for the user. The DOS prompt (also called a **system prompt**) will always display the **default drive** or the logged-on drive. The terms are synonymous. On a floppy drive system, after booting the computer with a diskette in drive A:, the DOS prompt will appear as

```
A>
```

On a hard-disk system, if booted with no disk in drive A:, the DOS prompt would appear as

```
C>
```

What does this mean for the user? A command or file may be accessed from the default drive without using the drive designation. Two keystrokes are saved. The difference in keystrokes is demonstrated by the following two command lines. They have the same results if the default drive is C:.

```
FORMAT A:
C:FORMAT A:
```

By simply typing in the drive designation (only) at the DOS prompt, we can change the default or logged-on drive. If logged on to drive C:, the DOS prompt appears as

```
C>
```

By typing **B:** and pressing the <ENTER> Key, also referred to as the <RETURN> Key or carriage return, the prompt switches to

```
B>
```

THE PATH

As you begin to work with external DOS commands in later units, you will find that you can identify the **path** to the command and still execute it--even if the command is not in the current directory or drive. The PATH command and the concept of directories and subdirectories are covered later. For now, realize that a path can be

specified. It simply tells the operating system which "route" (or path) should be taken to find a command, filename, or data.

Do not be overly concerned with the path concept at this point. This is just one of many concepts introduced in this unit but explored more fully in later units.

THE COMPLETE FILENAME

A completely assembled filename might appear as

```
A:\LETTER3.DOC
```

This would be an appropriate filename for a letter created by a word processor using the .DOC extension which is stored in the root directory (indicated by the back-slash character) on drive A:.

In summary, the complete filename equals

[Drive Designation] + [Path] + Filename + [Extension]

Even though the above represents the file specification or complete filename, most users consider the file specification as synonymous with the term *filename*. The file on drive C: called DOCUMENT with a .DOC extension is usually referred to as DOCUMENT, not C:DOCUMENT.DOC.

WILDCARD CHARACTERS

Two characters are reserved for a special use and may not be used in the filename or extension. These characters are the question mark (?) and the asterisk (*). They are referred to as global **wildcard** characters or **wildcard characters**. Either term signifies that they may be substituted for one or more characters in a filename. The question mark substitutes for one character; the asterisk may substitute for one or more characters.

Depending on how we structure filenames, these wildcard characters may be very helpful. For example, it would be helpful when backing up specific files if we formed a habit of using common characters in files with common data.

Text files storing a group of letters might be named

LETTER1.DOC
LETTER2.DOC
LETTER3.DOC
LETTER4.DOC

When backing up the files to disk, all of the LETTER files could be copied with one command rather than several. The DOS command COPY will duplicate the

specified files. By using the ? wildcard, the following command would copy every file from drive A: to drive B: which began with LETTER and ended with a single character (any character, not just 1 through 9) and which had the .DOC extension:

```
COPY A:LETTER?.DOC B:
```

We will come back to wildcard characters in the Guided Activities of upcoming units. They are very handy little tools. For now, remember that

1. ? means any one character may occupy that position in a filename.

2. * means any characters may occupy that position and the remaining positions in the filename or extension.

3. The ? and * characters may not be used as part of the actual filename given a file.

DISK HANDLING GUIDELINES

Whether working with DOS, an application program, or data disks, the handling of floppy disks is crucial. Diskettes are the tools of the computing trade. If you fail to handle these tools properly, trouble and inconvenience will follow.

If a computer includes a hard-disk system, you still must work with floppy disks. The most obvious uses of diskettes are to **archive** (back up or make a spare copy of) application program disks and to back up data. You cannot avoid working with diskettes--so it is best to learn how to use rather than abuse them.

The surface of the diskette must be protected from any element or activity that might damage it. Damage can occur by exposing the surface to contaminants (dust, dirt, fingerprints, liquids), physical damage (folding, creasing, bending, heavy objects, indentations from writing tools), heat (direct and prolonged sunlight, or temperatures above 125 degrees F), cold (below 50 degrees F), and magnetic fields (small motors, stereo speakers, magnets, non-electronic ringing telephones, televisions).

Another kind of disk protection is necessary to prevent accidental erasure or modification of important files on your disk. It is called write-protection.

NOTE: The term write-protection indicates that the files on a disk may be read but not altered. A **write-protected** disk is the same as a **read-only** disk. The data and programs may be loaded or viewed but not modified. You cannot write data to a disk once the write-protection tab is properly placed on it. 5¼-in. diskettes are write-protected by adding a self-adhesive tape over the write-protection notch (upper right corner on the side with the label). 3½-in. diskettes are write-protected by shifting a small plastic slide to open the write-protect window (turn the diskette case over with the label edge toward you--the circular spindle will be showing--then push the plastic slide down to open the window). To avoid accidentally formatting a disk, overwriting important data, or erasing important files, always write-protect the disk.

The 5¼-in. disk is packaged differently from the newer 3½-in. disk. The 5¼-in. disk is typically sealed in a black plastic jacket with a window through which the disk drive read-write head accesses data. The easiest method of damaging a diskette is to contaminate the magnetic oxide surface through that window. Each 5¼-in. diskette comes with a paper envelope. To further protect the recording surface, the disk should always be housed in its envelope when not in use.

Figure 1-5 identifies the parts of a 5¼-in. diskette including the disk envelope, permanent protective jacket, and write-protection notch. To write-protect this type of disk, simply use the write-protection tab (a small piece of self-adhesive tape) to cover the write-protection notch.

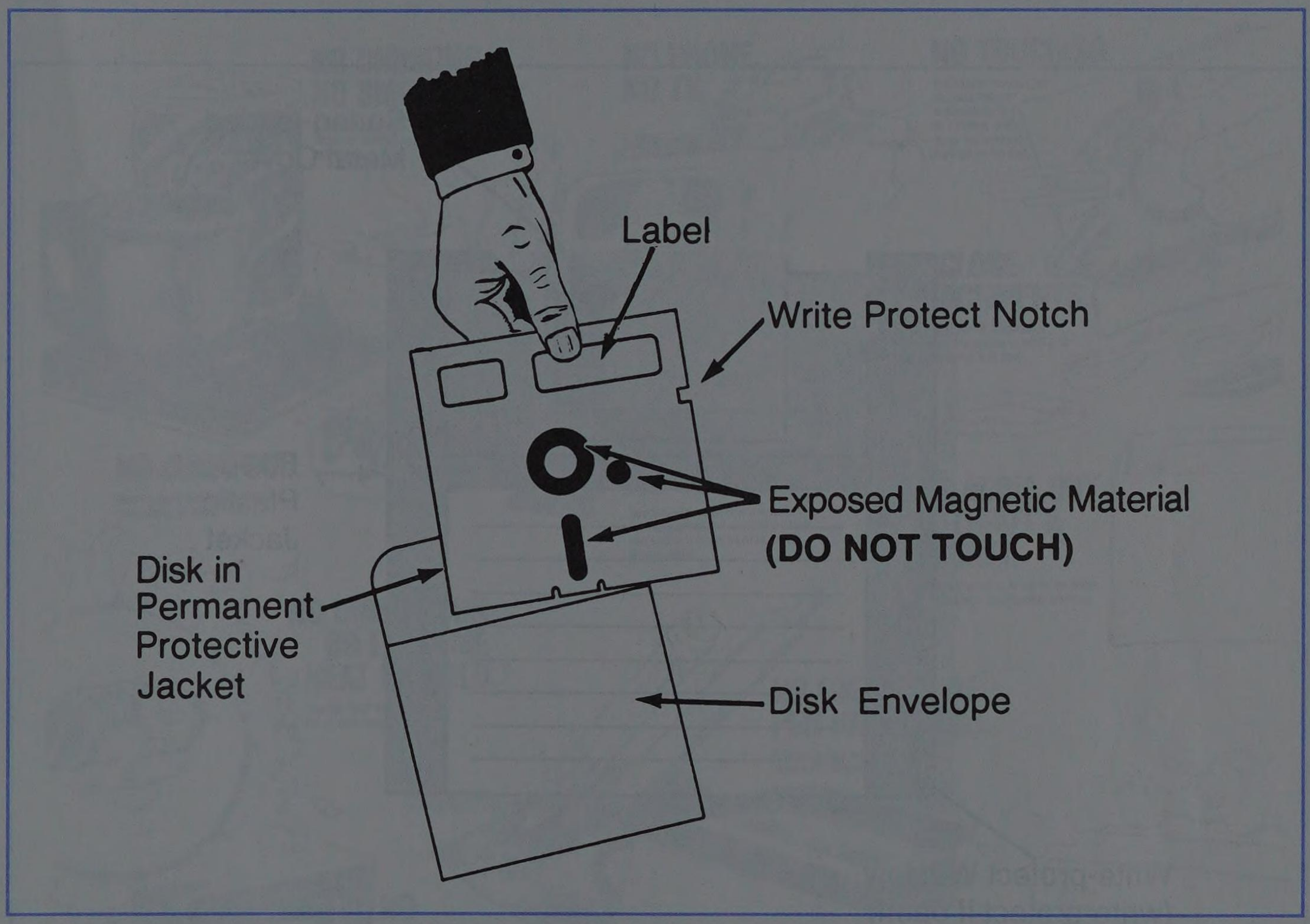


Fig. 1-5. Parts of the 5¼-in. floppy disk and proper holding position.

The 3½-in. disk comes in a permanent rigid plastic container. It includes a small plastic slide to write-protect the disk. The 3½-in. diskette is constructed with a spring-loaded metal cover that protects the disk access area from damage. The disk drive retracts the cover when the diskette is inserted in the drive. You should never open the cover manually because damage may occur to the recording surface. 3½-in. disks may come with a clear plastic envelope, but not always.

The parts of the 3½-in. disk are identified in Fig. 1-6.

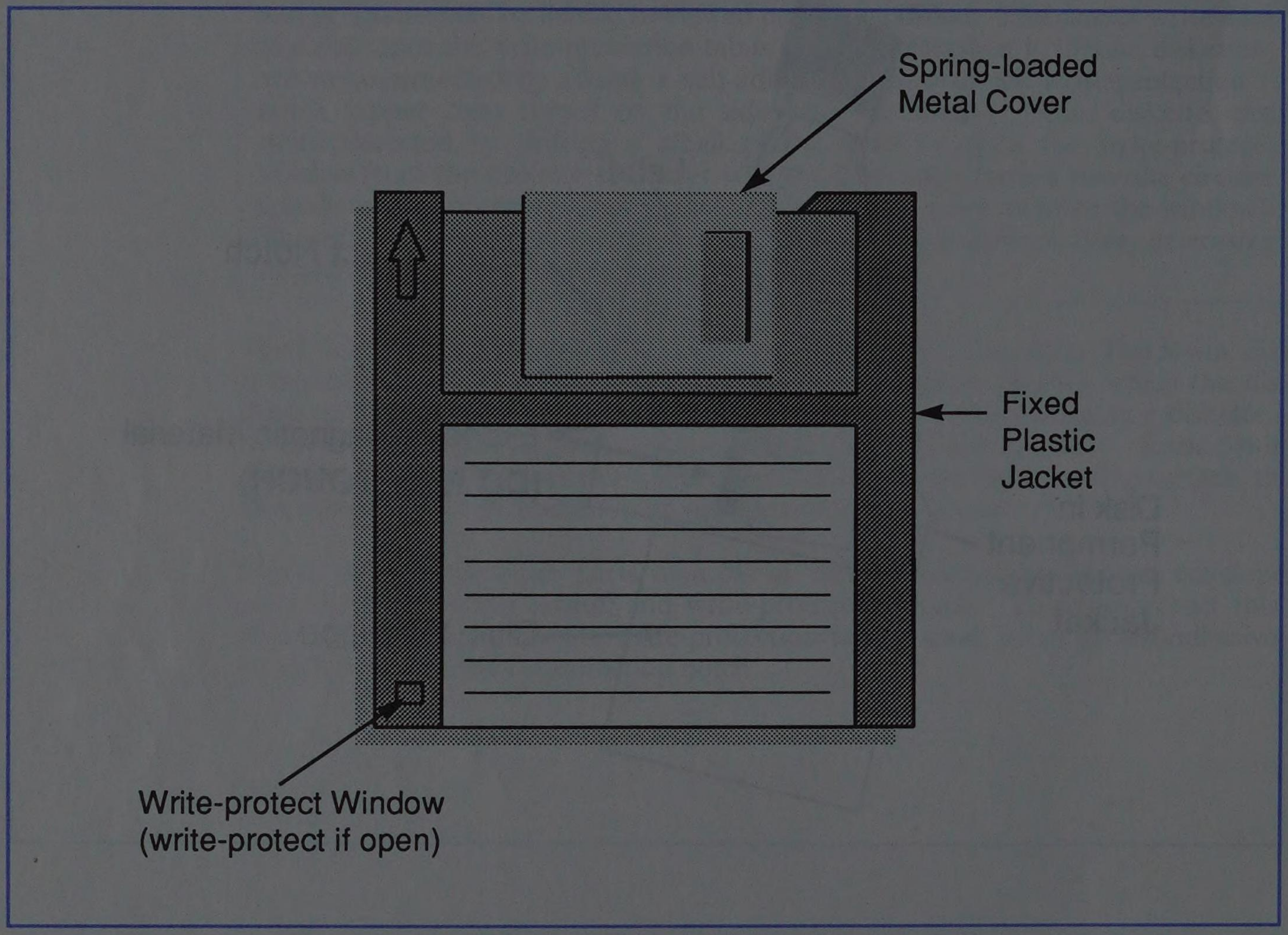


Fig. 1-6. Parts of the 3½-in. floppy disk.

Check Fig. 1-7 for a listing of important disk handling guidelines. No matter which type of disk you use, these rules still apply. Careful adherence to these guidelines will save you problems in the future.

Fig. 1-7. Disk handling guidelines.

BASIC ASSUMPTIONS

PC DOS/MS-DOS is a standard. While standards do exist, there is still phenomenal diversity when you look at the equipment currently installed in homes, schools, colleges, businesses, and industry. While we have attempted to make this text applicable to a wide variety of users, certain assumptions were made. If you understand the assumptions, you may be able to benefit from the text--even if your equipment is not the same as that listed below. The Guided Activities throughout this text are constructed while assuming a few facts about your computer system and the software needed. Those assumptions include

1. access to an IBM PC, XT, AT, Personal System/2 (PS/2), or compatible;

2. access to DOS version 3.x and its documentation (in order to answer the Documentation Research questions). The version of choice is PC DOS 3.3 or MS-DOS 3.3 and the *IBM Disk Operating System Version 3.3 Reference* manual, or the *Microsoft MS-DOS User's Guide and User's Reference* (for operating system 3.3) However, other versions of DOS 3.x and their accompanying documentation can also be used;

3. access to a system with at least one floppy disk drive and a hard disk.

Each Guided Activity will instruct you to boot your PC system. The next unit details the process and provides the basic knowledge you need to complete the activities in each unit.

A HISTORY OF DOS

If you are curious about the evolutionary history of MS-DOS/PC DOS, Appendix D provides additional information. In addition to a concise history, the appendix includes a chart showing the changes offered by each new version of DOS and a listing of internal and external DOS commands.

REVIEW QUESTIONS

1. What is the difference between the operating system on a mainframe and DOS on an IBM PC?

2. Define DOS.

3. Who developed IBM PC DOS and MS-DOS?

4. Where is IBM PC DOS stored for use by the computer?

5. Can application programs (i.e., word processors, spreadsheets, and data managers) be run without DOS?

6. What is a syntax diagram? Why is syntax important to a computer user?

7. When may internal and external DOS commands be executed?

8. What is the maximum number of characters that may be used in a filename? In an extension?

9. Name the two DOS wildcard characters. Can they be used when naming a file? Why or why not?

10. Which of the following are not legal filenames?

    LETTER\TO\MOTHER.TXT
    FINAL.LTR
    CALLHOME.DOC
    $%&9441!.BAK
    1.BAT
    COMMAND.COMM
    12345678.*90

11. How does the user change the default drive?

12. What is the difference between the default drive and the loggedon drive?

13. What is meant by the term case-sensitive? Is DOS case-sensitive when naming a file or executing a DOS command?

14. Why is NEWSNOTE.DOC a better filename than &$123NTE.DOC?

15. Explain what is meant by "booting your system".

16. What is the difference between the DOS prompt and the system prompt?

DOCUMENTATION RESEARCH

Documentation Research questions are included at the end of most units to encourage you to explore the documentation that accompanies DOS 3.x. All Documentation Research within the text uses the *IBM Disk Operating System Version 3.3 Reference* manual and the *Microsoft MS-DOS User's Guide and User's Reference* for version 3.3 (which is really two completely separate manuals in one softcover volume). However, documentation for earlier versions of DOS 3.x may also be used with minor modifications. For the following questions, read Chapters 1 and 2 in the *IBM Disk Operating System Version 3.3 Reference* manual or Chapter 1 in the *Microsoft MS-DOS User's Reference* and Chapter 2 in the *Microsoft MS-DOS User's Guide*.

For users of PC DOS:

1. What are the five types of diskettes supported by DOS 3.x? How much storage space, in kilobytes (K) or megabytes (MB), does each store?

2. What is drive compatibility? How does it affect the use of diskettes?

3. List the complete reserved names of all devices under DOS 3.x.

4. Give three examples from the reference manual, using both wildcard characters, for which the resulting files listed would be identical.

For users of MS-DOS:

1. What are the three types of disks listed in Chapter 2 of the *User's Guide*?

2. May reserved words which are invalid for filenames under DOS 3.x. be used as file extensions?

3. What is the wildcard abbreviation, and what does it represent?

PART II

FUNDAMENTAL DOS COMMANDS

This second part of the text covers the fundamental disk operating system commands. These commands are routinely used by PC users in the course of their daily and weekly work. Even if the user has an aversion to programming and the more complex types of computer tasks, these basic commands are essential.

The fundamental DOS commands fall into four groups: disk preparation commands, housekeeping commands, file management commands, and organizational commands. Such tasks as preparing a disk for use; copying, deleting, and renaming files; comparing files and disks; and organizing files into subdirectories are covered in Units 2, 3, 4, and 5.

UNIT 2

DISK PREPARATION COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. explain the use of the FORMAT command;
2. format a non-system diskette;
3. format a non-system diskette with a volume label;
4. format a system diskette;
5. label a formatted diskette;
6. transfer the DOS files with SYS command;
7. create a duplicate diskette with the DISKCOPY command;
8. compare a newly copied diskette with the original.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. DISKCOMP
2. DISKCOPY
3. SYS
4. LABEL
5. FORMAT.

ASSIGNMENTS

The assignments to be completed for this unit are

____Guided Activity: Computer Terms Dictionary (Appendix A): **FAT.**
____Guided Activity: Change the System Prompt.
____Guided Activity: Change to the DOS Subdirectory.
____Guided Activity: Format a Non-System Diskette.
____Guided Activity: Format a Non-System Diskette with a Volume Label.
____Guided Activity: Format a System Diskette.
____Guided Activity: Format a System Diskette with a Volume Label.
____Guided Activity: Transfer the DOS System Files with the SYS Command.
____Guided Activity: Add a Volume Label to a Diskette.
____Guided Activity: Make a Duplicate Diskette with the DISKCOPY Command.
____Guided Activity: Compare Two Diskettes with the DISKCOMP Command.
____Review Questions.
____Unit Exercises.
____Documentation Research.

Place a check in front of those assigned by your instructor.

DISK PREPARATION

A quick review of the basic construction of a diskette and a hard disk will enable you to have a better understanding of the preparation commands. The FORMAT, LABEL, DISKCOPY, DISKCOMP, and SYS commands are necessary commands to allow you to work and store data on your disks.

DISK CONSTRUCTION

Disks are divided into two categories: floppy and hard disks. The disks used with a microcomputer can be either 5¼-in. and/or 3½-in. disks, depending on the disk drives installed within the computer. It is now possible to have both a 3½-in. disk drive and a 5¼-in. disk drive installed in a personal computer.

An unused floppy or hard disk cannot store information until it is prepared with the DOS commands. The construction of these disks and the amount of information they are capable of storing vary, depending on the disk size and construction, the mechanical construction of the disk drive it is used in, and the DOS command used to prepare the disk for data storage.

5¼-INCH DISK

A 5¼-in. disk is made of a flexible plastic material coated with metallic oxide and covered with a semi-rigid cardboard disk jacket or envelope. When holding a 5¼-in. disk, the label should be on the top right side of the disk with the write-protective

notch on the left side of the disk. The head access slot where information from the computer is transferred to and from the disk should be farthest away from you. It is important not to touch the shiny disk surface showing in the head access slot as this can disable the transference of information to and from the computer and the disk. As you hold the disk by the label end, the write-protective notch is located on the left side and keeps information from being written to the disk and deleted from the disk. When the notch is covered by a write-protect tab, no information may be transferred to or deleted from the disk by the computer.

3½-INCH DISK

A 3½-in. disk is composed of the same material as a 5¼-in. disk but has a hard plastic outside cover. The disk has a metal slide covering the head access slot to protect the disk when it is not in the computer. A 3½-in. disk also has a write-protective notch, but it is different from the write-protective notch on the 5¼-in. disk. The 3½-in. disk has a plastic slide for the write-protective notch. When the slide is moved to the position that shows an opening in the notch, the disk is protected, and the computer is incapable of writing information to and deleting information from the disk. The construction of the 3½-in. disk makes it more durable than the 5¼-in. disk and less susceptible to damage.

WATCH THOSE DISKETTES

A 1.2MB high density 5¼-in. drive can read or write data to a formatted low density 5¼-in. disk.

If you format a high density diskette in a 5¼-in. 1.2mb high density drive using a /4 parameter, data may be written to or read from the high density diskette using a low density drive.

If you format a 5¼-in. diskette in a high density disk drive without a /4 parameter, you will only be able to read and write data to that diskette in a high density drive.

A 1.44MB high density 3½-in. drive can read and write data to a 3½-in. 720K diskette, but a 720K 3½-in. drive cannot read or write data to a formatted 1.44MB high density diskette.

HARD DISKS

A hard disk is a non-removable disk which is enclosed and designed to reside permanently in the computer. The actual disk is made of metal platters coated with the same material as the floppy disk. The number of platters and the size of platters determine the amount of the storage capacity on the disk. The hard disk transfers information through the read/write heads of the disk drive. The read/write heads of a hard disk are more precise than those of a floppy disk drive and are more susceptible to damage from shock and misuse.

TRACKS AND SECTORS

An unformatted disk does not contain tracks or sectors and is incapable of storing information. The process of formatting a disk places concentric magnetic tracks on the disk which contain sectors where bytes of information are stored. Each sector on a disk can contain 512 bytes of information. The magnetic structures of both hard and floppy disks are similar with regard to tracks and sectors. The difference between the hard disk and the floppy are due to the number of tracks on the disk and the number of sides available for each track. The sides of a disk are named with either a 1 or a 0. The top side of a disk (the side with the label) is considered to be side 1, and the bottom side of the disk is considered to be side 0. Each side of the disk has the same number of tracks and sectors for storing magnetic information.

The following table shows the numbers of bytes available for files on various disks:

Table 1

Disk Size	360K	720K	1.2MB	1.44MB	20MB
Total Bytes	362,496	730,112	1,213,952	1,457,664	21,149,696*
Storage Bytes	368,640	737,280	122,800	1,474,560	21,204,992

* formatted with DOS 3.3 System files.

LOADING DOS 3.30

1. If you are using a computer with a hard disk drive, make sure you do not have a diskette in drive A: before you turn on the computer.

2. If the PC is off, turn it on. This activity will boot your computer system and load DOS. DOS must be loaded before any computer activity can take place. This method is called a *cold start* or *cold boot*.

3. If the PC is on, reset the system by holding down the CTRL and ALT Keys and simultaneously pressing the DEL Key. This method of booting is called a *warm start* or *warm boot*.

NOTE: Systems with internal clocks and a clock driver do not prompt for the date and time.

You will see the disk drive light come on while DOS is being loaded, and you may hear some disk drive noise. When DOS is loaded and ready, you will see the screen response

```
Current date is Tue 1-01-1980
Enter new date (mm-dd-yy):_
```

When entering the new date, use the form MM/DD/YY, where MM=Month, DD=Day, and YY=Year. In the example 8/21/90, "8" represents the month of August, "21" is the day of the month, and "90" represents the year 1990.

4. Enter today's date.

5. Press <ENTER>.

After entering the date, you can enter the time. Time is displayed in hours, minutes, seconds, and hundredths of seconds. DOS displays the following screen response for entering time.

```
Current time is 0:00:41.24
Enter new time:_
```

DOS 3.X only understands time entered in 24-hour time. By typing 18:30, you enter a time of 6:30 P.M.

6. Enter current time.

7. Press <ENTER>.

The last message you receive from DOS is

```
The IBM Personal Computer DOS
Version 3.30 (C)Copyright International Business Machines Corp 1981,1987
(C)Copyright Microsoft Corp 1981, 1986

C>_
```

GUIDED ACTIVITY

CHANGE THE SYSTEM PROMPT: The PROMPT command can be used to identify the current DOS path and also to change the system prompt. To easily identify the current subdirectory, type **PROMPT pg** and press <ENTER>. The PROMPT command will be studied in detail in Unit 3.

1. Type **PROMPT pg**

2. Press <ENTER>.

GUIDED ACTIVITY

CHANGE TO THE DOS SUBDIRECTORY: You will need to change to the subdirectory that contains the DOS files. To change to a subdirectory, type **CD** followed by the subdirectory name. For example, if the DOS files are in a subdirectory called \DOS, type **CD\DOS** and press <ENTER>.

1. Type **CD\DOS**
2. Press <ENTER>.

FORMAT COMMAND

The FORMAT command is one of the most used DOS file commands. Before you can use a diskette, whether it is a 5¼-in. disk, 3½-in. disk, or hard disk, you must prepare it to receive data. Formatting a disk creates a clean surface on the disk and removes all data. It checks the disk surface for errors in construction and for the suitability of the disk to receive tracks and sectors. When bad sectors are found on the disk surface, they are identified so that data will not be stored in these areas. The directory that you view with the DIR command is created with the FORMAT command. Initializing, or formatting, creates a *FAT* (file allocation table) which allows you to see the filenames, size, and the times, and dates the files were created or last modified and stored on the disk. After the disk is formatted, you can determine the number of usable bytes available for storage and the number of bytes on the disk. You can create different disk conditions by using parameters or switches. Switches or parameters are composed of a slash (/) followed by one of several characters. In the following list, the "d:" after the FORMAT command is the drive designator for the drive that contains the disk to be formatted.

COMMAND	ACTION
`FORMAT d:`	Format a non-system 9-or-15 sector-per-track diskette.
`FORMAT d:/V`	Format non-system diskette with a Volume Label.
`FORMAT d:/S`	Format a diskette with the DOS operating system. This diskette can then be used to boot the system at start-up. This procedure places the system files IBMBIO.COM, IBMDOS.COM, and COMMAND.COM on the diskette. The MS-DOS equivalent files are IO.SYS, MSDOS.SYS, and COMMAND.COM.
`FORMAT d:/4`	Format a non-system high-capacity disk to be used in a double-sided 360K drive.
FORMAT d:/N:9/T:80	Format a non-system high-capacity disk to be used in a 720K drive.

#1

EVERYDAY FORMAT COMMAND SWITCHES

In your work with the FORMAT command, you will find that you use some switches more than others. The switches in this section are the ones that are most often needed in your everyday DOS work.

GUIDED ACTIVITY

FORMAT A NON-SYSTEM DISKETTE: The FORMAT d: command (where d: designates the drive for the disk to be formatted) creates a formatted non-system diskette. A non-system diskette is not self-booting and is used to store data. In this activity a diskette will be formatted using a 360K drive.

WARNING: Be very careful when you use the FORMAT command with a hard disk. If you do not identify a target drive with the FORMAT command, the computer will format the current drive. In the case of a hard drive, that would be disastrous. All data would be removed from the hard-disk drive.

1. Type **FORMAT A:**
2. Press <ENTER>.
3. Insert a new diskette in drive A:.
4. Press <ENTER>.
5. If you do not want to format another disk, type **N**
6. Press <ENTER>.

If the diskette surface has any defects, DOS will not store data at those locations. The number of bytes in bad sectors will be reported by DOS after formatting a diskette.

NOTE: Turn on your printer. To create your own picture of the FORMAT command results that you have just completed, press SHIFT-PRTSC. This procedure is called a *screen dump*, and will output the contents of the computer screen to the printer.

Figure 2-1 shows the screen response after formatting a diskette.

```
C:\DOS>FORMAT A:
Insert new diskette for drive A:
and strike ENTER when ready

Format complete

    362496 bytes total disk space
    362496 bytes available on disk

Format another (Y/N)?N
C:\DOS>
```

Fig. 2-1. Formatting a non-system diskette.

> **NOTE:** DIR is an internal DOS command that is used to scan the files on your disk.

7. Type **DIR A:**

8. Press <ENTER>.

Figure 2-2 shows the root directory of a formatted disk after using the FORMAT command without any switches. Notice there are no files visible on the disk.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

File not found

C:\DOS>
```

Fig. 2-2. Root directory of a formatted disk.

GUIDED ACTIVITY

FORMAT A NON-SYSTEM DISKETTE WITH A VOLUME LABEL: The FORMAT d:/V command (where d: designates the drive for the disk to be formatted) creates a non-system diskette with a volume label. The /V switch will allow you to place the volume label MEMO DISK on the diskette. If you use the DIR command with this disk, you will see the MEMO DISK label on the diskette.

1. Type **FORMAT A:/V**
2. Press <ENTER>.
3. Insert a diskette in drive A:.
4. Press <ENTER>.
5. When prompted for "Volume label," type **MEMO DISK**
6. Press <ENTER>.
7. If you do not want to format another disk, type **N**

8. Press <ENTER>.

Figure 2-3 shows the screen results when formatting with a volume label.

```
C:\DOS>FORMAT A:/V
Insert new diskette for drive A:
and strike ENTER when ready

Format complete

Volume label (11 characters, ENTER for none)? MEMO DISK


    362496 bytes total disk space
    362496 bytes available on disk

Format another (Y/N)?N
C:\DOS>
```

Fig. 2-3. Formatting a disk with a volume label.

9. Type **DIR A:**

10. Press <ENTER> to see the volume label.

Figure 2-4 shows the volume label produced on the disk.

```
C:\DOS>DIR A:

 Volume in drive A is MEMO DISK
 Directory of  A:\

File not found

C:\DOS>
```

Fig. 2-4. Directory of a volume label.

GUIDED ACTIVITY

FORMAT A SYSTEM DISKETTE: The `FORMAT d:/S` command (where d: designates the drive for the disk to be formatted) creates a self-booting system diskette. The /S switch makes a diskette self-booting by copying the DOS system files IBMBIO.COM, IBMDOS.COM, and COMMAND.COM to the diskette.

1. Type **`FORMAT A:/S`**
2. Press <ENTER>.
3. Insert a diskette in drive A:.
4. Press <ENTER>.
5. If you do not want to format another disk, type **N**
6. Press <ENTER>.

Figure 2-5 shows the screen response when formatting a system diskette.

```
C:\DOS>FORMAT A:/S
Insert new diskette for drive A:
and strike ENTER when ready

Format complete
System transferred

    362496 bytes total disk space
     78848 bytes used by system
    283648 bytes available on disk

Format another (Y/N)?N
C:\DOS>
```

Fig. 2-5. Formatting a system diskette.

7. Type **DIR**
8. Press <ENTER>.

Figure 2-6 shows the COMMAND.COM file on the system disk.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

COMMAND  COM     25307   3-17-87  12:00p
       1 File(s)    283648 bytes free

C:\DOS>
```

Fig. 2-6. Directory of a system disk showing the DOS file COMMAND.COM.

GUIDED ACTIVITY

FORMAT A SYSTEM DISKETTE WITH A VOLUME LABEL:

1. Type **FORMAT A:/S/V**
2. Press <ENTER>.
3. Insert a diskette in drive A:.
4. Press <ENTER>.
5. When prompted for a "Volume label," type **CLASS DISK**
6. Press <ENTER>.
7. If you do not want to format another disk, type **N**
8. Press <ENTER>.

Figure 2-7 shows the screen response when formatting a system diskette with a volume label.

```
C:\DOS>FORMAT A:/S/V
Insert new diskette for drive A:
and strike ENTER when ready

Format complete
System transferred

Volume label (11 characters, ENTER for none)? CLASS DISK


    362496 bytes total disk space
     78848 bytes used by system
    283648 bytes available on disk

Format another (Y/N)?N
C:\DOS>
```

Fig. 2-7. Formatting a system diskette with a volume label.

9. Type **Dir A:**

10. Press <ENTER> to see the volume label.

Figure 2-8 shows the volume label displayed on the screen along with the file COMMAND.COM.

```
C:\DOS>DIR A:

 Volume in drive A is CLASS DISK
 Directory of  A:\

COMMAND  COM     25307   3-17-87  12:00p
       1 File(s)    283648 bytes free

C:\DOS>
```

Fig. 2-8. Directory of system diskette with a volume label.

SYS COMMAND

The SYS command is used to copy the DOS hidden files from a DOS system diskette.

GUIDED ACTIVITY

TRANSFER THE DOS SYSTEM FILES WITH SYS THE COMMAND: The only response yow will receive from DOS will be a "System transferred" message. The DOS hidden files cannot be listed with the DOS DIR command.

1. Type **SYS A:**
2. Press <ENTER>.

Figure 2-9 shows the screen results of the SYS command.

```
C:\DOS>SYS A:
System transferred

C:\DOS>
```

Fig. 2-9. SYS command.

3. Type **DIR A:**

4. Press <ENTER>.

Figure 2-10 shows that there are no files visible on the disk.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

File not found

C:\DOS>
```

Fig. 2-10. View of the disk in drive A:.

LABEL COMMAND

The LABEL command is used to add or change a volume label on a formatted disk.

GUIDED ACTIVITY

ADD A VOLUME LABEL TO A DISKETTE:

1. Type **LABEL A:**
2. Press <ENTER>.
3. Type **HIDDEN**
4. Press <ENTER>.

Figure 2-11 shows the results of the LABEL command.

```
C:\DOS>LABEL A:

Volume in drive A has no label

Volume label (11 characters, ENTER for none)? HIDDEN


C:\DOS>
```

Fig. 2-11. View of the disk labeled HIDDEN.

5. Type **DIR A:**
6. Press <ENTER>.

Figure 2-12 shows the disk with the volume label HIDDEN.

```
C:\DOS>DIR A:

 Volume in drive A is HIDDEN
 Directory of  A:\

File not found

C:\DOS>
```

Fig. 2-12. View of the volume label HIDDEN.

DISKCOPY COMMAND

The DISKCOPY command is used to make duplicate copies of a disk.

GUIDED ACTIVITY

MAKE A DUPLICATE DISK WITH THE DISKCOPY COMMAND: In this activity you will create a duplicate disk using the DISKCOPY command on one drive.

1. Type **DISKCOPY A: A:**
2. Press <ENTER>.
3. Insert the source diskette (the disk you want to copy) in drive A:.
4. Press <ENTER>.
5. When requested, insert your target diskette in drive A:. This is the diskette that will became the duplicate of the original diskette in drive A:.

6. Press <ENTER>.

7. If you do not want to copy another disk, type **N**

Figure 2-13 shows the screen response with the DISKCOPY command. On a previously formatted diskette, you will not receive the message "Formatting while copying".

```
C:\DOS>DISKCOPY A: A:

Insert SOURCE diskette in drive A:

Press any key when ready . . .

Copying 40 tracks
9 Sectors/Track, 2 Side(s)

Insert TARGET diskette in drive A:

Press any key when ready . . .


Copy another diskette (Y/N)?N

C:\DOS>
```

Fig. 2-13. DISKCOPY command.

8. Type **DIR**

9. Press <ENTER>.

Figure 2-14 shows a view of the newly created disk after using the DISKCOPY command.

```
C:\DOS>DIR A:

 Volume in drive A is HIDDEN
 Directory of  A:\

File not found

C:\DOS>
```

Fig. 2-14. View of a newly created disk.

10. Type **CLS**

11. Press <ENTER>.

DISKCOMP COMMAND

The DISKCOMP command is used to compare two diskettes. It lets you know if the duplicate diskette you made with DISKCOPY is the same as the original. DISKCOMP is only valid immediately after using the DISKCOPY command. DISKCOMP cannot be used with the COPY command.

GUIDED ACTIVITY

COMPARE TWO DISKETTES WITH THE DISKCOMP COMMAND:

1. Type **DISKCOMP A: A:**

2. Press <ENTER>.

3. Insert the original diskette in drive A:.
4. Press <ENTER>.
5. When requested, insert the duplicate diskette in drive A:.
6. Press <ENTER>.
7. If you do not want to compare another disk, type **N**

Figure 2-15 shows the screen response to the DISKCOMP command.

```
C:\DOS>DISKCOMP A: A:

Insert FIRST diskette in drive A:

Press any key when ready . . .

Comparing 40 tracks
9 sectors per track, 2 side(s)

Insert SECOND diskette in drive A:

Press any key when ready . . .

Compare OK

Compare another diskette (Y/N) ?N

C:\DOS>
```

Fig. 2-15. DISKCOMP command.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have covered the common forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`DISKCOMP A: A:`	Compares a copied diskette with the original.
`DISKCOPY A: A:`	Creates a duplicate diskette using one drive.
`FORMAT A:`	Creates a formatted non-system diskette.
`FORMAT A:/V`	Creates a formatted non-system diskette with a volume label.
`FORMAT A:/S`	Creates a self-booting system diskette.
`FORMAT A:/S/V`	Creates a self-booting system diskette with a volume label.
`SYS A:`	Copies DOS hidden system files to the disk in drive A:.

REVIEW QUESTIONS

1. When do you need to use the FORMAT command with the /S switch?
2. What procedure would you follow to copy the DOS system files to a disk?
3. How many bytes are used to format a system diskette?
4. What procedure would you follow to make a duplicate diskette?
5. What procedure would you follow to check that the duplicate disk is the same as the original?

UNIT EXERCISES

1. Format a disk as a DOS system disk with a volume label. Use your last name for the volume label. Turn on your printer. Make a screen dump (press <SHIFT-PRTSC> to screen dump) of the results of the exercise to a printer. Hand in the results of the screen dump to your instructor.

2. Use the LABEL command to change the volume label on the system diskette. Your instructor will tell you the new volume label name. Screen dump the results, and hand them in to your instructor.

3. Make a duplicate of the diskette you made in Exercise 2.

DOCUMENTATION RESEARCH

1. Research the *IBM Disk Operating System Version 3.3 Reference* manual for DOS commands and parameters not used in class.

2. How would you make a DOS system disk using the SYS command?

APPLICATION A

HOUSEKEEPING COMMANDS

PART 1

FORMATTING A BLANK DISKETTE:

A. Format a blank diskette without any parameters.

PART 2

CREATING A LABEL WITH THE LABEL COMMAND:

A. Use the diskette formatted in PART 1-A, and add your name as a label.

B. Make a duplicate of the disk with your name.

UNIT 3

HOUSEKEEPING COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. use the PROMPT command to change the system prompt;
2. use the CLS command to clear the monitor screen;
3. use the DATE command to change or update the current date;
4. use the TIME command to change or update the current time;
5. use the DIR command to determine the date, time, and bytes used by a file;
6. use the CHKDSK command to determine total amount of disk space in bytes;
7. use the CHKDSK command to determine bytes available on a disk;
8. use the CHKDSK command to determine the amount of memory available in the computer;
9. use the CHKDSK command to determine the amount of memory used by the computer;
10. use the CHKDSK command with the /V parameter to view files and their paths on the disk you are checking;
11. use the CHKDSK command with the /F parameter to repair errors on your diskette;
12. use the CHKDSK/F command on a hard disk.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. CHKDSK
2. CLS
3. DATE
4. DIR
5. PROMPT
6. TIME.

ASSIGNMENTS

The assignments to be completed for this unit are

_____Guided Activity: Clear the Screen with the CLS Command.
_____Guided Activity: Use the PROMPT Command to Change Your System PROMPT.
_____Guided Activity: View the Directory with the System Prompt.
_____Guided Activity: Use the DATE Command to Change the System Date.
_____Guided Activity: Use the TIME Command to Change the System Time.
_____Guided Activity: Use the DIR Command to Scan a File.
_____Guided Activity: Use <CTRL-S> to Stop the Screen from Scrolling.
_____Guided Activity: Use the DIR Command to Produce Horizontal Filenames.
_____Guided Activity: Use the DIR Command with a Pause.
_____Guided Activity: Check a Disk Using the CHKDSK Command.
_____Guided Activity: Check a Disk Using CHKDSK with the /V Parameter.
_____Guided Activity: Check a Disk Using CHKDSK with the /F Parameter.
_____Guided Activity: Check a Hard Disk Using CHKDSK with the /F Parameter.
_____Review Questions.
_____Unit Exercises.
_____Documentation Research.

Place a check in front of those assigned by your instructor.

HOUSEKEEPING COMMANDS

Housekeeping command is the name given to a command that comes into play regularly. Though you might not use the commands all the time, the results of these commands are felt every time you use DOS. You will now learn the six housekeeping commands: CHKDSK, DIR, CLS, PROMPT, DATE, and TIME.

1. Turn on the computer.

2. Enter the date and time.

3. Type **`PROMPT $p$g`**

4. Press <ENTER>.

CLS COMMAND

The CLS command is used to clear the screen of unwanted characters or text.

GUIDED ACTIVITY

CLEAR THE SCREEN WITH THE CLS COMMAND:

1. Type **ABCDEF** at C:\>.

2. Press <ENTER>.

Figure 3-1 shows the screen response to a bad command. The letters ABCDEF are not a command or filename, so DOS gives a "Bad command or file name" message.

```
C:\>ABCDEF
Bad command or file name

C:\>
```

Fig. 3-1. Bad command.

3. Type **CLS**

4. Press <ENTER>.

Now the only item on the screen will be the system prompt.

PROMPT COMMAND

The PROMPT command is a fun and powerful tool for changing your DOS PROMPT. The simple command called PROMPT, followed by text, can change the system prompt. If you typed **PROMPT HI!** and pressed <ENTER>, you would be greeted with "HI!" on the screen.

There are special meta-strings that can be used with the PROMPT text. These characters are preceded by $. In our example above, if you wanted to have the drive identifier as a > (greater-than sign) character following HI! you would type HI! ng. The $n is for the default drive, and the $g is for the greater-than sign.

These characters, preceded by the dollar sign, will cause the system prompt to produce the following on the screen:

PROMPT meta-strings

$b	| (vertical bar)
$d	the DOS date
$e	escape character
$g	> (greater-than sign)
$h	backspace
$l	< (less-than sign)
$n	default drive
$p	Directory of the default drive
$q	= (equal sign)
$t	DOS time
$v	DOS version number
$$	dollar sign
$_	underscore; causes the start of a new line

GUIDED ACTIVITY

USE THE PROMPT COMMAND TO CHANGE YOUR SYSTEM PROMPT: The PROMPT command can be used to change your DOS prompt and make some interesting greetings when you start your computer.

1. Type **PROMPT HI!**

2. Press <ENTER>.

Figure 3-2 shows the change to the system prompt.

```
C:\>PROMPT HI!

HI!
```

Fig. 3-2. HI! system prompt.

3. Type **CLS**
4. Press <ENTER>.
5. Type **PROMPT**
6. Press <ENTER>.

Figure 3-3 shows the screen response to the PROMPT command.

7. Type **CLS**
8. Press <ENTER>.

```
HI!PROMPT

C>
```

Fig. 3-3. PROMPT.

GUIDED ACTIVITY

VIEW THE DIRECTORY WITH THE SYSTEM PROMPT: The PROMPT command can be used to change your DOS prompt so you are able to determine the subdirectory you are using.

1. Type **PROMPT pg**
2. Press <ENTER>.

Figure 3-4 shows that the screen response to the PROMPT command is C:\>. The back-slash represents the root directory. When you use the PROMPT command shown in step 1, the root directory will always be shown in this position.

3. Type **CLS**
4. Press <ENTER>.

```
C>PROMPT $p$g

C:\>
```

Fig. 3-4. C:\> system prompt.

DATE COMMAND

The DATE command can be used either to enter the current date or to change the current system date. It is an internal DOS command that does not require the DOS utility files to be resident in the default disk drive.

GUIDED ACTIVITY

USE THE DATE COMMAND TO CHANGE THE SYSTEM DATE:

1. Type **DATE**
2. Press <ENTER>.

Figure 3-5 shows the screen response to the DATE command.

In this example, you are going to enter the date May 1, 1989. The configuration for the DATE command you are to enter is (mm-dd-yy) or (mm-dd-yyyy). The month (mm) can be entered as 05 or 5, the day (dd) can be entered as 01 or 1, and the year (yy), (yyyy) can be entered as a two or four-digit number (in the example, the year

could be 1989 or just 89). You may use a period, slash, or hyphen to separate the date numbers.

```
C:\>DATE
Current date is Wed  1-25-1989
Enter new date (mm-dd-yy):
```

Fig. 3-5. DATE command.

3. Type **01.01.1989**
4. Press <ENTER>.
5. Type **DATE**
6. Press <ENTER>.

Let's enter an incorrect date and see the DOS response!

7. Type **13-25-79**
8. Press <ENTER>.

DOS will respond with "Invalid date" and request you to enter a new date. There were two errors in the above date entry. The first is obvious: there are only 12 months in a year, so 13 is not correct. The second error is not so obvious as the

first. DOS will allow us to enter years only from 1980 through 1999. The year 1979 was incorrect, and DOS responded with the "invalid date" message.

9. Type **01/25/89**

10. Press <ENTER>.

Figure 3-6 show the series of DATE command entries.

```
C:\>DATE
Current date is Wed  1-25-1989
Enter new date (mm-dd-yy): 01.01.1989

C:\>DATE
Current date is Sun  1-01-1989
Enter new date (mm-dd-yy): 13-25-79

Invalid date
Enter new date (mm-dd-yy): 01/25/89

C:\>
```

Fig. 3-6. DATE command entries.

11. Type **CLS**

12. Press <ENTER>.

13. Type **DATE**

14. Press <ENTER>.

15. Enter today's date.

16. Press <ENTER>.

17. Type **CLS**

18. Press <ENTER>.

TIME COMMAND

The TIME command will allow you to enter or change the system time that you want to appear on the files on the disk.

GUIDED ACTIVITY

USE THE TIME COMMAND TO CHANGE THE SYSTEM TIME:

1. Type **TIME**

2. Press <ENTER>.

Figure 3-7 shows that DOS will respond with the current time and ask you to enter a new time. For example, the time 6:00 P.M. would be entered as 18:00. Remember to separate the hours and minutes with a colon.

```
C:\>TIME
Current time is  9:26:01.20
Enter new time:
```

Fig. 3-7. TIME command.

3. Enter the current time.
4. Press <ENTER>.
5. Type **CLS**
6. Press <ENTER>.

DIR COMMAND

The DIR command is used to view files on your disks. When this command is used, the files on your disk are listed down the screen until all the files have been scanned by DOS. Using the file COMMAND.COM as an example, the DIR command shows us the following file on the screen:

COMMAND COM 25307 3-17-87 12:00p

COMMAND is the filename.
COM is the file extension.
25,307 bytes have been used.
3-17-87 is the date the file was created or last updated.
12:00p is the time the file was created or last updated.

GUIDED ACTIVITY

USE THE DIR COMMAND TO SCAN A FILE:

> **NOTE:** Remember, the command CD\ stands for change to the root directory of the default disk.

1. Type **CD\DOS**
2. Press <ENTER> to change from the root directory C:\ to the subdirectory C:\DOS.
3. Type **DIR**
4. Press <ENTER> to view the DOS files.

Figure 3-8 shows the end of the DOS file list.

```
MORE     COM       313   3-17-87   12:00p
NLSFUNC  EXE      3060   3-17-87   12:00p
PRINT    COM      9026   3-17-87   12:00p
PRINTER  SYS     13590   3-17-87   12:00p
RECOVER  COM      4299   3-18-87   12:00p
REPLACE  EXE     11775   3-17-87   12:00p
RESTORE  COM     34643   3-17-87   12:00p
SELECT   COM      4163   3-17-87   12:00p
SHARE    EXE      8608   3-17-87   12:00p
SORT     EXE      1977   3-17-87   12:00p
SUBST    EXE      9909   3-17-87   12:00p
SYS      COM      4766   3-17-87   12:00p
TREE     COM      3571   3-17-87   12:00p
VDISK    SYS      3455   3-17-87   12:00p
XCOPY    EXE     11247   3-17-87   12:00p
EGA      CPI     49065   3-18-87   12:00p
LCD      CPI     10752   3-17-87   12:00p
4201     CPI     17089   3-18-87   12:00p
5202     CPI       459   3-17-87   12:00p
BASIC    PIF       369   3-17-87   12:00p
BASICA   PIF       369   3-17-87   12:00p
MORTGAGE BAS      6251   3-17-87   12:00p
       52 File(s)   2058240 bytes free

C:\DOS>
```

Fig. 3-8. Scanning a file with DIR.

GUIDED ACTIVITY

USE <CTRL-S> TO STOP THE SCREEN FROM SCROLLING: To stop the screen from scrolling, simultaneously press the <CTRL> Key and the <S> Key, and then release. The screen will stop, and you can view these filenames. To start the screen scrolling again, press any key.

1. Type **DIR**
2. Press <ENTER>.
3. Press <CTRL-S> to stop the screen from scrolling.
4. Press <ENTER> to start the screen scrolling again.
5. Type **CLS**
6. Press <ENTER>.

GUIDED ACTIVITY

USE THE DIR COMMAND TO PRODUCE HORIZONTAL FILENAMES: The DIR command with the /W parameter will cause the filenames to appear horizontally across the screen.

1. Type **DIR/W**

2. Press <ENTER>.

Figure 3-9 shows the screen response to the DIR/W command.

```
C:\DOS>DIR/W

 Volume in drive C has no label
 Directory of  C:\DOS

.              ..             COMMAND  COM    ANSI     SYS    APPEND   EXE
ASSIGN   COM   ATTRIB  EXE   BACKUP   COM    BASIC    COM    BASICA   COM
CHKDSK   COM   COMP    COM   COUNTRY  SYS    DEBUG    COM    DISKCOMP COM
DISKCOPY COM   DISPLAY SYS   DRIVER   SYS    EDLIN    COM    FASTOPEN EXE
FDISK    COM   FIND    EXE   FORMAT   COM    GRAFTABL COM    GRAPHICS COM
JOIN     EXE   KEYB    COM   KEYBOARD SYS    LABEL    COM    MODE     COM
MORE     COM   NLSFUNC EXE   PRINT    COM    PRINTER  SYS    RECOVER  COM
REPLACE  EXE   RESTORE COM   SELECT   COM    SHARE    EXE    SORT     EXE
SUBST    EXE   SYS     COM   TREE     COM    VDISK    SYS    XCOPY    EXE
EGA      CPI   LCD     CPI   4201     CPI    5202     CPI    BASIC    PIF
BASICA   PIF   MORTGAGE BAS
       52 File(s)   2058240 bytes free

C:\DOS>
```

Fig. 3-9. DIR/W command.

GUIDED ACTIVITY

USE THE DIR COMMAND WITH A PAUSE: The DIR command with the /P parameter will cause the screen to pause from scrolling after 23 files are on the screen.

1. Type **DIR/P**

2. Press <ENTER>.

The `DIR/P` command will cause the screen to scroll through 23 lines of text at one time. Once 23 lines are displayed, the screen stops scrolling and waits for any key to be pressed before continuing.

3. Press <ENTER>.

CHKDSK COMMAND

The CHKDSK command is used to check a disk's directories, files, and allocation table. It also tells you the total amount of disk space in bytes, bytes available, memory available, and memory used.

GUIDED ACTIVITY

CHECK A DISK USING THE CHKDSK COMMAND: For this activity you will need a formatted system disk in drive A:.

> **WARNING:** Make sure you identify the drive that contains the disk you are going to format; otherwise, you may format the hard disk.

1. Type **`FORMAT A:/S`**
2. Press <ENTER>.
3. Insert new diskette in drive A:.
4. Press <ENTER>.
5. To not format another disk, type **`N`**
6. Press <ENTER>.
7. Type **`CLS`**
8. Press <ENTER>.
9. Type **`CHKDSK A:`**
10. Press <ENTER>.

Figure 3-10 shows the screen response to the CHKDSK command. In this activity, the disk in drive A: was formatted with a /S parameter. Because you used this parameter, 53,248 bytes are used for the hidden DOS files. The amount of total memory or bytes free may vary, depending on the amount of memory in your computer.

```
C:\DOS>CHKDSK A:

  362496 bytes total disk space
   53248 bytes in 2 hidden files
   25600 bytes in 1 user files
  283648 bytes available on disk

  655360 bytes total memory
  434176 bytes free

C:\DOS>
```

Fig. 3-10. CHKDSK command.

11. Type **CLS**

12. Press <ENTER>.

GUIDED ACTIVITY

CHECK A DISK USING CHKDSK WITH THE /V PARAMETER: The CHKDSK command with the /V parameter lets you view the files and their paths on the disk you are checking. If you look at the disk files using the /V parameter, you can see the DOS hidden files.

In the following example, you are going to check a disk that has been formatted with the /S parameter.

1. Type **CHKDSK A:/V**

2. Press <ENTER>.

Figure 3-11 shows the screen response to the CHKDSK A:/V command. The files listed on the screen are the DOS system files.

```
C:\DOS>CHKDSK A:/V
Directory A:\
      A:\IBMBIO.COM
      A:\IBMDOS.COM
      A:\COMMAND.COM

   362496 bytes total disk space
    53248 bytes in 2 hidden files
    25600 bytes in 1 user files
   283648 bytes available on disk

   655360 bytes total memory
   434176 bytes free

C:\DOS>
```

Fig. 3-11. CHKDSK A:/V command on a formatted system disk.

3. Type **CLS**
4. Press <ENTER>.

GUIDED ACTIVITY

CHECK A DISK USING CHKDSK WITH THE /F PARAMETER: The CHKDSK command with the /F parameter lets DOS repair errors in the directory or file allocation table on your disk. If DOS finds errors on your disk, DOS will respond with that information and give you the option of repairing the errors. The /F parameter causes DOS to group into a file any lost clusters of data it finds on the disk being checked. Lost clusters are allocation units that contain data but are not associated with the file allocation table.

In the following example, you are going to check a disk that has been formatted with the /S parameter.

1. Type **CHKDSK A:/F**
2. Press <ENTER>.

Figure 3-12 shows the screen response to the CHKDSK A:/F command.

```
C:\DOS>CHKDSK A:/F

  362496 bytes total disk space
   53248 bytes in 2 hidden files
   25600 bytes in 1 user files
  283648 bytes available on disk

  655360 bytes total memory
  434176 bytes free

C:\DOS>
```

Fig. 3-12. CHKDSK A:/F command on a formatted system disk.

3. Type **CLS**
4. Press <ENTER>.

GUIDED ACTIVITY

CHECK A HARD DISK USING CHKDSK WITH THE /F PARAMETER:

1. Type **CHKDSK/F**
2. Press <ENTER>.

Figure 3-13 shows the response to the CHKDSK/F command with a 20MB hard disk.

3. If you have any lost clusters of data, and want to convert them to a file, type **Y** and press <ENTER> when DOS makes the request. If you are not requested to convert clusters, DOS does not have data for you to convert.

```
C:\DOS>CHKDSK/F

2 lost clusters found in 1 chains.
Convert lost chains to files  (Y/N)? Y

 21030912 bytes total disk space
    55296 bytes in 3 hidden files
    69632 bytes in 29 directories
 18823168 bytes in 733 user files
     4096 bytes in 1 recovered files
    20480 bytes in bad sectors
  2058240 bytes available on disk

   655360 bytes total memory
   434176 bytes free

C:\DOS>
```

Fig. 3-13. CHKDSK/F command with a hard disk.

4. Type **CLS**

5. Press <ENTER>.

Lost cluster files are stored on the root directory of the hard disk. The files start with the name FILE0000.CHK. In the next step you will look at the root directory for any files that were created by CHKDSK. If you did not create files, the next step will show "File not found."

6. Type **DIR C:\FILE*.CHK**

7. Press <ENTER>.

Figure 3-14 shows the file FILE0000.CHK in the root directory.

8. Type **CLS**

9. Press <ENTER>.

```
C:\DOS>DIR C:\FILE*.CHK

 Volume in drive C has no label
 Directory of  C:\

FILE0000 CHK     4096   5-16-89   9:40a
       1 File(s)   2058240 bytes free

C:\DOS>
```

Fig. 3-14. CHKDSK FILE0000.CHK in the root directory.

In the next step you will use the DOS TYPE command to show the contents of a file created with CHKDSK/F. Since the file was created from lost clusters of data, your file may be different from the one in the book.

> **NOTE:** The TYPE command is an internal DOS command that displays the contents of a file on the computer screen. It will display to the screen exactly what is in the file. The command syntax for the TYPE command is straightforward. Use the command name, tell DOS where the file is, and list the filename.

10. Type **TYPE C:\FILE0000.CHK**
11. Press <ENTER>.

Figure 3-15 shows the data in the file FILE0000.CHK.

```
C:\>TYPE FILE0000.CHK
```

Fig. 3-15. FILE0000.CHK.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`CHKDSK`	Checks the general condition of the diskette.
`CHKDSK /F`	Checks the diskette and allows you to correct errors.
`CHKDSK /V`	Allows you to view the files and paths on a disk.
`DATE`	Allows you to enter or change the current date.
`DIR`	Lists files on the current disk drive.
`DIR/P`	Causes the screen to stop scrolling after 23 files are viewed.
`TIME`	Allows you to enter or change the current time.

REVIEW QUESTIONS

1. How would you view the hidden DOS files?
2. What procedure would you follow to determine the memory in your computer?
3. What command will tell you when a file was created or last used?
4. What command will allow you to view your disk files and their paths?
5. What command will clear your monitor of unwanted characters?
6. What command will tell you if you have any bad bytes on your disk?

UNIT EXERCISES

1. Format a disk as system disk, and use the CHKDSK command to view the hidden DOS files. Make a screen dump of the screen and hand the results in to your instructor.
2. Create a system prompt that displays the date, time, subdirectory, and disk drive. Make a screen dump of the screen, and hand the results in to your instructor.

DOCUMENTATION RESEARCH

1. Research the *IBM Disk Operating System Version 3.3 Reference* manual for DOS information about diskette construction.

UNIT 4

FILE MANAGEMENT COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. use the COPY command to copy files to your screen;
2. use the COPY command to copy files to your printer;
3. use the COPY command to copy files to other disk drives;
4. create a file using the COPY command;
5. copy files using wildcard characters;
6. combine files to form one file;
7. use the COMP command to compare files;
8. use the RENAME command to change the name of a file;
9. output a file to the screen with the TYPE command;
10. use the DELETE command to delete files.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. COPY
2. COMP
3. DELETE
4. RENAME
5. TYPE.

ASSIGNMENTS

The assignments to be completed for this unit are

_____Review and understand the following computer terms listed in the
_____Computer Terms Dictionary (Appendix A): **End-of-File Mark, Wildcard, Internal, and External Commands.**
_____Guided Activity: Start the Computer.
_____Guided Activity: Create a File Using the COPY Command.
_____Guided Activity: Copy a File to the Screen.
_____Guided Activity: Copy a File to Another Drive.
_____Guided Activity: Create another File.
_____Guided Activity: Copy a File with Wildcard Characters.
_____Guided Activity: Copy a File to the Same Drive and Directory.
_____Guided Activity: Copy More Files with Wildcards.
_____Guided Activity: Copy with *.*.
_____Guided Activity: Combine Two Files.
_____Guided Activity: Copy a File to the Printer.
_____Guided Activity: Compare Two Files with the COMP Command.
_____Guided Activity: Rename a File.
_____Guided Activity: Output to Screen with the TYPE Command.
_____Guided Activity: Delete a Single File from the Current Drive.
_____Guided Activity: Delete a Group of Files with Wildcard Characters.
_____Review Questions.
_____Unit Exercises.
_____Documentation Research.

Place a check in front of those assigned by your instructor.

FILE MANAGEMENT COMMANDS

File management commands will help you manage the files on your floppy and hard disks. The COPY, DELETE, and RENAME commands are internal DOS commands and are not required to be on the default disk drive after you have loaded DOS into memory. The COMP command is an external DOS command which will require the COMP.COM file to be resident on the default drive.

Managing your disks will require you to copy, delete, rename, and compare files within your disks and subdirectories. The commands that will accomplish these tasks, and that will be covered in this unit, are COPY, COMP, DELETE, RENAME, and TYPE.

GUIDED ACTIVITY

START THE COMPUTER:

1. Turn on the computer.

2. Enter the current date and time.

In this unit you will need to create a subdirectory for the guided activities. The name of the subdirectory that will be used in this unit is \EXER. The subdirectory \EXER can be made on a floppy disk or a hard disk. Subdirectories will be discussed in length in Unit 5.

3. To create the subdirectory \EXER, type **MD\EXER**

4. Press <ENTER>.

5. To move to the subdirectory \EXER, type **CD\EXER**

6. Press <ENTER>.

7. Type **CLS**

8. Press <ENTER>.

9. Place a formatted disk in drive A:.

COPY COMMAND

The COPY command is an internal DOS command used to copy a file or files to a disk from another disk or subdirectory. The COPY command will also copy input from your keyboard and display that input to your monitor and printer.

GUIDED ACTIVITY

CREATE A FILE USING THE COPY COMMAND: You can create a file directly from your keyboard with the COPY command. When you do this, you must tell DOS the file is being input from the keyboard. You accomplish this by typing the word CON after the command COPY and before the filename. If you want to create a file called CLASS1.LST, the command would be

```
COPY CON CLASS1.LST
```

If you want the file to be stored on a drive other than the default drive, use a drive identifier in the command syntax after the word CON. The command to store the file on drive A: would look like this

```
COPY CON A:CLASS1.LST
```

This procedure creates an ASCII text file which you will be able to copy from disk to disk, see on your screen, and print on your printer.

When you are typing in COPY CON, you can make corrections only on the current

line. If you find you have made a mistake and wish to cancel the file, press <CTRL-C>.

1. Type **COPY CON CLASS1.LST**
2. Press <ENTER>.
3. Type **Dean Swenson**
4. Press <ENTER>.
5. Type **Bill Richards**
6. Press <ENTER>.
7. Type **Hank Miller**
8. Press <ENTER>.

When you press <CTRL-Z> or press function key <F6>, a special character is placed at the end of the file. When DOS comes to the end of your file, it will know the file is finished because of this character. The character is called an **End-of-file mark**.

9. Press <CTRL-Z>.
10. Press <ENTER>.

Figure 4-1 shows the correct file structure for creating the file. Notice that after you pressed the <ENTER> Key, the statement "1 Files(s) copied" appeared on the screen.

```
C:\EXER>COPY CON CLASS1.LST
Dean Swenson
Bill Richards
Hank Miller
^Z
        1 File(s) copied

C:\EXER>
```

Fig. 4-1. CLASS1.LST file.

11. Type **DIR \EXER**
12. Press <ENTER>.

Figure 4-2 shows the CLASS1.LST file on the default disk drive.

```
C:\EXER>COPY CON CLASS1.LST
Dean Swenson
Bill Richards
Hank Miller
^Z
        1 File(s) copied

C:\EXER>DIR \EXER

 Volume in drive C has no label
 Directory of  C:\EXER

.            <DIR>       9-01-89   2:39a
..           <DIR>       9-01-89   2:39a
CLASS1   LST         42   9-01-89   2:40a
        3 File(s)   1222656 bytes free

C:\EXER>
```

Fig. 4-2. CLASS1.LST file on the default disk drive.

13. Type **CLS**

14. Press <ENTER>.

GUIDED ACTIVITY

COPY A FILE TO THE SCREEN: You can use the COPY command to copy the contents of a file to the screen. To see the results of the last activity, we again use COPY CON, but this time we place CON after the filename in your command. The command syntax is `COPY CLASS1.LST CON` to output the file contents to the screen.

1. Type **COPY CLASS1.LST CON**

2. Press <ENTER>.

Figure 4-3 shows the screen response to the COPY CLASS1.LST CON command.

```
C:\EXER>COPY CLASS1.LST CON
Dean Swenson
Bill Richards
Hank Miller
        1 File(s) copied

C:\EXER>
```

Fig. 4-3. Copying CLASS1.LST to the screen.

3. Type **CLS**
4. Press <ENTER>.

GUIDED ACTIVITY

COPY A FILE TO ANOTHER DRIVE: In this activity you are going to copy the file named CLASS1.LST from drive C:\EXER, the source drive, to drive A:, the target drive. In order to assure that the file is copied correctly, you are going to use the /V (verify) parameter with the COPY command. This parameter will verify that the files copied to the new drive are correct and that the file is intact. The command syntax to COPY a file with the /V parameter to another drive is

```
COPY C:\EXER\CLASS1.LST A:/V.
```

This command will copy the file CLASS1.LST from drive C:\EXER to drive A: and verify that the file was copied correctly.

1. Type **COPY C:\EXER\CLASS1.LST A:/V**
2. Press <ENTER>.
3. Type **DIR A:**
4. Press <ENTER> to view the file on drive A:.
5. Type **CLS**
6. Press <ENTER>.

GUIDED ACTIVITY

CREATE ANOTHER FILE: This time you will save the copied file directly to drive A: rather than drive C:\EXER. Notice how the syntax changes in this procedure: `COPY CON A:CLASS2.LST`.

1. Type **COPY CON A:CLASS2.LST**
2. Press <ENTER>.
3. Type **Joe Pine**
4. Press <ENTER>.
5. Type **Mary Paul**
6. Press <ENTER>.
7. Type **Susan Donald**
8. Press <ENTER>.
9. Press <CTRL-Z>.
10. Press <ENTER>.

Figure 4-4 shows the file structure of the CLASS2.LST file on drive A:.

```
C:\EXER>COPY CON A:CLASS2.LST
Joe Pine
Mary Paul
Susan Donald
^Z
        1 File(s) copied

C:\EXER>
```

Fig. 4-4. CLASS2.LST file.

11. Type **CLS**
12. Press <ENTER>.
13. To view the files on drive A:, type **DIR A:**
14. Press <ENTER>.

Figure 4-5 shows the files on drive A:.

```
C:\EXER>DIR A:

 Volume in drive A has no label
 Directory of  A:\

CLASS1   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
       2 File(s)    360448 bytes free

C:\EXER>
```

Fig. 4-5. Directory of drive A:.

15. Type **CLS**

16. Press <ENTER>.

GUIDED ACTIVITY

COPY A FILE WITH WILDCARD CHARACTERS: You can use wildcard characters to copy files to another disk or subdirectory. The asterisk (*) takes the place of any filename or extension in the file. The asterisk (*) is a wildcard that can be used to copy both of the files on drive A: to drive C:\EXER. In our example, both files have the same extension LST, but different filenames. We can substitute the wildcard asterisk in place of the filenames and type only the extension. The syntax would look like this: `COPY A:*.LST C:\EXER`.

1. Type **COPY A:*.LST C:\EXER**

2. Press <ENTER>.

3. Type **DIR C:\EXER*.LST**

4. Press <ENTER> to view the files.

Figure 4-6 shows the use of the asterisk wildcard in the copying of files.

```
C:\EXER>COPY A:*.LST C:\EXER
A:CLASS1.LST
A:CLASS2.LST
        2 File(s) copied

C:\EXER>DIR C:\EXER\*.LST

 Volume in drive C has no label
 Directory of  C:\EXER

CLASS1   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
        2 File(s)   1220608 bytes free

C:\EXER>
```

Fig. 4-6. Asterisk wildcard used in filenames.

5. Type **CLS**

6. Press <ENTER>.

GUIDED ACTIVITY

COPY A FILE TO THE SAME DRIVE AND DIRECTORY: DOS will not allow you to copy a file to the same disk drive subdirectory in which the file resides. If you want to do this, you must give the file a different name. You are going to give the file called CLASS1.LST a new name while using the COPY command in this activity.

1. Type **COPY CLASS1.LST CLASS3.LST**

2. Press <ENTER>.

3. Type **DIR C:\EXER*.LST**

4. Press <ENTER>.

Figure 4-7 shows the result of using the COPY command to copy a file to the same drive and subdirectory. You now have two files that are the same, in the same place, but with different names.

```
C:\EXER>COPY CLASS1.LST CLASS3.LST
        1 File(s) copied

C:\EXER>DIR C:\EXER\*.LST

 Volume in drive C has no label
 Directory of  C:\EXER

CLASS1   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
CLASS3   LST        42   9-01-89   2:40a
        3 File(s)   1218560 bytes free

C:\EXER>
```

Fig. 4-7. Copy and change the name of a file.

5. Type **CLS**

6. Press <ENTER>.

GUIDED ACTIVITY

COPY MORE FILES WITH WILDCARDS: The question mark (?) is a wildcard character that takes the place a character in a filename or extension. The asterisk takes the place of any filename or extension in a file. When you request the COPY command to copy the files in this example, it reads the files as if they were all the same and copies them.

In this guided activity the question mark (?) will take the place of the number one in the filename CLASS1.LST and the asterisk (*) will take the place of the extension LST.

1. Type **COPY CLASS?.* A:**
2. Press <ENTER>.

Figure 4-8 shows the results of copying files with a question mark and an asterisk.

```
C:\EXER>COPY CLASS?.* A:
CLASS1.LST
CLASS2.LST
CLASS3.LST
        3 File(s) copied

C:\EXER>
```

Fig. 4-8. Copy a file with a question mark and an asterisk.

3. Type **CLS**
4. Press <ENTER>.

GUIDED ACTIVITY

COPY WITH *.*: In this activity you will copy the files by using wildcard characters in both the filename and the extension. The asterisks substitute for any filenames and any extensions.

1. Type **COPY A:*.* C:\EXER**
2. Press <ENTER>.

Figure 4-9 displays the fast method of copying files using the asterisk (*) global wildcard characters.

```
C:\EXER>COPY A:*.* C:\EXER
A:CLASS1.LST
A:CLASS2.LST
A:CLASS3.LST
        3 File(s) copied

C:\EXER>
```

Fig. 4-9. Global wildcard characters.

3. Type **CLS**
4. Press <ENTER>.

GUIDED ACTIVITY

COMBINE TWO FILES: The COPY command will allow you to combine files. The files are combined by placing a plus sign between the files and then giving a new filename for the combined file. In this activity you will combine two files into a new file called TOTCLASS.ROS. The command syntax to combine the files CLASS1.LST and CLASS2.LST is

```
COPY CLASS1.LST+CLASS2.LST TOTCLASS.ROS.
```

1. Type **COPY CLASS1.LST+CLASS2.LST TOTCLASS.ROS**
2. Press <ENTER>.

Figure 4-10 shows the screen response when combining two files into one. Notice that the new file is not on the screen, but it has been made because you can see the "1 File(s)copied" message.

```
C:\EXER>COPY CLASS1.LST+CLASS2.LST TOTCLASS.ROS
CLASS1.LST
CLASS2.LST
        1 File(s) copied

C:\EXER>
```

Fig. 4-10. Combining two files into one.

3. Type **CLS**
4. Press <ENTER>.
5. Type **DIR \EXER**
6. Press <ENTER> to view the files.

Figure 4-11 shows the four files that you have created.

```
C:\EXER>DIR \EXER

 Volume in drive C has no label
 Directory of  C:\EXER

.            <DIR>          9-01-89   2:39a
..           <DIR>          9-01-89   2:39a
CLASS1   LST              42   9-01-89   2:40a
CLASS2   LST              35   9-01-89   2:48a
CLASS3   LST              42   9-01-89   2:40a
TOTCLASS ROS              78   9-01-89   2:56a
        6 File(s)    1216512 bytes free

C:\EXER>
```

Fig. 4-11. Four created files.

7. Type **CLS**

8. Press <ENTER>.

GUIDED ACTIVITY

COPY A FILE TO THE PRINTER: The COPY command will allow you to copy the contents of a file to the printer. The command `COPY TOTCLASS.ROS PRN` will copy the contents of the file TOTCLASS.ROS to the printer PRN.

1. Turn on your printer.

2. Type **COPY TOTCLASS.ROS PRN**

3. Press <ENTER>.

Figure 4-12 shows the screen response "1 File(s)copied" when you output the contents of the file to the printer.

```
C:\EXER>COPY TOTCLASS.ROS PRN
        1 File(s) copied

C:\EXER>
```

Fig. 4-12. Copying a file to a printer.

4. Type **CLS**
5. Press <ENTER>.

The output to your printer will list all the names that have been combined into the one file.

Dean Swenson
Bill Richards
Hank Miller
Joe Pine
Mary Paul
Susan Donald

COMP COMMAND

The COMP command is used to compare the contents of a file, or a like group of files, to a like file, or group of files. You use the COMP command after the COPY command to ensure that the files you copied are identical to the originals.

GUIDED ACTIVITY

COMPARE TWO FILES WITH THE COMP COMMAND: The COMP command is an external DOS command. It is used to compare the contents of a file that has been copied with the COPY command. In this activity you will compare the CLASS1.LST file on the hard disk and with the CLASS1.LST file on the disk in drive A:.

1. Place the disk that contains the file CLASS1.LST in drive A:.

> **NOTE:** Remember the COMP command is an external DOS command and you must have the COMP.COM file in the current subdirectory in order to execute the COMP command. The COMP.COM file is in the \DOS subdirectory.

2. Type **CD\DOS**
3. Press <ENTER>.
4. Type **COMP C:\EXER\CLASS1.LST A:CLASS1.LST**
5. Press <ENTER>.
6. If you do not want to compare more files, type **N**

Figure 4-13 shows the screen response to the COMP command.

```
C:\EXER>CD\DOS

C:\DOS>COMP C:\EXER\CLASS1.LST A:CLASS1.LST


C:\EXER\CLASS1.LST and A:CLASS1.LST

Eof mark not found

Files compare ok

Compare more files (Y/N)? N

C:\DOS>
```

Fig. 4-13. COMP command.

7. Type **CLS**
8. Press <ENTER>.

RENAME COMMAND

The RENAME (REN) command is an internal DOS command which does not require the DOS files to be resident in the computer. The REN command allows you to change the name of your file. As you saw in the COPY command, changing a filename allows you to copy the file to the same drive and subdirectory.

GUIDED ACTIVITY

RENAME A FILE: To change the name of a file, you first type the name of the command, followed by the drive name, the old file name, and then the new file name. To rename the file CLASS1.LST to OFFICE.LST on disk drive A:, you would use this command: `REN A:CLASS1.LST OFFICE.LST`. Whenever you are working within the same drive and directory, you may omit the drive name. That command would look like this: `REN CLASS1.LST OFFICE.LST`. When

you invoke the command, the only response will be that the drive light will come on when the new name is written to the disk.

1. Type **DIR A:**
2. Press <ENTER> to display the files on drive A:.

Figure 4-14 shows a list the current files that are on the disk in drive A:.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

CLASS1   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
CLASS3   LST        42   9-01-89   2:40a
        3 File(s)    359424 bytes free

C:\DOS>
```

Fig. 4-14. List of files on drive A:.

3. Type **REN A:CLASS1.LST OFFICE.LST**
4. Press <ENTER>.
5. Type **DIR A:**
6. Press <ENTER> to see the file name change.

Figure 4-15 shows the file name change from CLASS1.LST to OFFICE.LST.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

CLASS1   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
CLASS3   LST        42   9-01-89   2:40a
        3 File(s)    359424 bytes free

C:\DOS>REN A:CLASS1.LST OFFICE.LST

C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

OFFICE   LST        42   9-01-89   2:40a
CLASS2   LST        35   9-01-89   2:48a
CLASS3   LST        42   9-01-89   2:40a
        3 File(s)    359424 bytes free

C:\DOS>
```

Fig. 4-15. REN command.

7. Type **CLS**
8. Press <ENTER>.
9. Type **REN A:CLASS2.LST HOME.LST**
10. Press <ENTER>.
11. Type **DIR A:**
12. Press <ENTER>.

Figure 4-16 shows the change you made with the REN command.

```
C:\DOS>REN A:CLASS2.LST HOME.LST

C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

OFFICE   LST        42   9-01-89   2:40a
HOME     LST        35   9-01-89   2:48a
CLASS3   LST        42   9-01-89   2:40a
        3 File(s)    359424 bytes free

C:\DOS>
```

Fig. 4-16. REN command.

13. Type **CLS**

14. Press <ENTER>.

TYPE COMMAND

The TYPE command displays the contents of a file to the computer screen. It will display to the screen exactly what is in the file. The command syntax for the TYPE command is straightforward. Use the command name, tell DOS where the file is, and list the filename. The command structure for the TYPE command is

```
TYPE A:HOME.LST
```

and will cause DOS to look to drive A: for the file HOME.LST and display it on the screen.

GUIDED ACTIVITY

OUTPUT TO SCREEN WITH THE TYPE COMMAND:

1. Type **TYPE A:HOME.LST**
2. Press <ENTER>.

Figure 4-17 shows the screen response for the TYPE command.

```
C:\DOS>TYPE A:HOME.LST
Joe Pine
Mary Paul
Susan Donald

C:\DOS>
```

Fig. 4-17. TYPE command.

3. Type **CLS**
4. Press <ENTER>.

DELETE/ERASE COMMAND

The DELETE and the ERASE commands both erase files from a disk. It does not matter which command you use; they both do the same thing. The advantage of the DELETE command is that it requires you to type fewer characters in the command name. The command syntax to delete the file CLASS3.LST file from the disk on drive A: is

```
DEL A:CLASS3.LST
```

Because the DELETE command removes files from your disk, you should be careful with the command. You can delete files one at a time, but by using the global wildcards you may also delete multiple files with a single command.

GUIDED ACTIVITY

DELETE A SINGLE FILE FROM THE CURRENT DRIVE: When you invoke the DEL command, you won't see the changes until you look at the directory of the disk. Again, the only visible response to the command will be that the drive light will momentarily come on while the file is being deleted.

Before files are deleted, check the drive for the files to be deleted.

1. Type **DIR A:**
2. Press <ENTER>.
3. Type **DEL A:CLASS3.LST**
4. Press <ENTER>.
5. Type **DIR A:**
6. Press <ENTER>.

Figure 4-18 shows the response to the command to delete a single file from drive A:.

```
C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

OFFICE   LST        42   9-01-89   2:40a
HOME     LST        35   9-01-89   3:20a
CLASS3   LST        42   9-01-89   2:40a
       3 File(s)    359424 bytes free

C:\DOS>DEL A:CLASS3.LST

C:\DOS>DIR A:

 Volume in drive A has no label
 Directory of  A:\

OFFICE   LST        42   9-01-89   2:40a
HOME     LST        35   9-01-89   3:20a
       2 File(s)    360448 bytes free

C:\DOS>
```

Fig. 4-18. Deleting a file.

7. Press **CLS**

8. Press <ENTER>.

GUIDED ACTIVITY

DELETE A GROUP OF FILES WITH WILDCARD CHARACTERS: The DEL command will allow the substitution of wildcard characters in the same way as the COPY command. To delete all the files with extensions named LST in the subdirectory \EXER type

```
DEL C:\EXER\*.LST
```

To delete all the files in the subdirectory \EXER on disk drive C:\, type

```
DEL C:\EXER\*.*.
```

1. Change to the subdirectory \EXER, type **CD\EXER**

2 Press <ENTER>.

3. Type **DIR C:\EXER*.LST**
4. Press <ENTER>.
5. To delete files with the extensions named LST, type **DEL C:\EXER*.LST**
6. Press <ENTER>.
7. Type **DIR C:\EXER*.LST**
8. Press <ENTER> to view the \EXER subdirectory on drive C:.

Figure 4-19 shows the response to the command to delete multiple files from a subdirectory.

```
C:\DOS>CD\EXER

C:\EXER>DIR C:\EXER\*.LST

 Volume in drive C has no label
 Directory of  C:\EXER

CLASS1   LST        42   9-01-89   5:37a
CLASS2   LST        35   9-01-89   5:42a
CLASS3   LST        42   9-01-89   5:37a
        3 File(s)   1218560 bytes free

C:\EXER>DEL C:\EXER\*.LST

C:\EXER>DIR C:\EXER\*.LST

 Volume in drive C has no label
 Directory of  C:\EXER

File not found

C:\EXER>
```

Fig. 4-19. Deleting files from a subdirectory.

9. Type **CLS**
10. Press <ENTER>.

11. Type **DIR A:*.***
12. Press <ENTER>.
13. To delete all the files on drive A:, type **DEL A:*.***
14. Press <ENTER>.

DOS responds with the message "Are you sure (Y/N)? _". If you type **Y** for yes and press <ENTER>, all the files on the disk in drive A: will be deleted.

15. Type **Y**
16. Press <ENTER>.
17. Type **DIR A:**
18. Press <ENTER>.

Figure 4-20 shows the response to the command to delete files with wildcards.

```
C:\EXER>DIR A:*.*

 Volume in drive A has no label
 Directory of  A:\

OFFICE   LST        42   9-01-89   5:37a
HOME     LST        35   9-01-89   5:42a
        2 File(s)    360448 bytes free

C:\EXER>DEL A:*.*
Are you sure (Y/N)?Y

C:\EXER>DIR A:

 Volume in drive A has no label
 Directory of  A:\

File not found

C:\EXER>
```

Fig. 4-20. Deleting all files with the *.* command.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`COMP A:*.* C:`	Compares all files on drive A: with the files having the same names on drive C:.
`COPY *.* A:`	Copies all files in the resident drive to drive A:.
`COPY C:\EXER\EXAMPLE.TXT A:`	Copies the file EXAMPLE.TXT from C:\EXER to drive A:.
`COPY C:\EXER\EXAMPLE.TXT A:/V`	Copies and verifies that the file was copied correctly.
`DEL A:*.*`	Deletes all files on drive A:.
`DEL A:EXAMPLE.TXT`	Deletes the file EXAMPLE.TXT on drive A:.
`REN A:JOE.TXT BILL.TXT`	Renames the file JOE.TXT on drive A: to BILL.TXT.
TYPE A:HOME.LST	Output the file HOME.LST to the screen.

REVIEW QUESTIONS

1. What command sequence would you use to create a file?
2. How would you copy a file to your printer?
3. How would you create a duplicate file on the same disk and subdirectory?
4. How would you combine two files?
5. How would you copy all the files on a disk to another disk with one command?
6. How would you delete all the files on a disk with one command?

UNIT EXERCISES

1. Use COPY CON to create a list of five people in a file called LIST.TXT.

2. Copy the file LIST.TXT to a file called NEWLST.TXT, and screen dump the screen to a printer. Hand the screen dump in to your instructor.

3. Rename the file LIST.TXT to your last name, and screen dump the screen to a printer. Hand the screen dump in to your instructor.

APPLICATION B

PART 1

FORMATTING A BLANK DISKETTE: In this unit, you will need a blank diskette to store your files.

A. Format a blank diskette without any parameters.

PART 2

CREATING TEXT FILES WITH COPY CON: In Unit 2, you learned how to build text files with the COPY CON command. In this application, you will build a file for each month of the year, combine files, and output the results to the printer.

A. Use the first three letters of the month for the filename and 88 for the extension. Place the name of the month within the appropriate file.

Make all months as follows :

JAN.88
FEB.88
MAR.88
APR.88
MAY.88
JUN.88
JUL.88
AUG.88
SEPT.88
OCT.88
NOV.88
DEC.88

Save these files for later use.

B. Use the DIR command, output the months to the printer. Hand the list in to your instructor.

PART 3

COMBINING FILES :

A. Combine the months into four quarters using the names QTR1, QTR2, QTR3, and QTR4, and use the DIR command to list the four QTR files on the printer. Hand the list in to your instructor.

B. Use the COPY command to copy the QTR files to the printer. Hand the results in to your instructor.

PART 4

RENAMING OF FILES:

A. Use the RENAME command to rename each file to

Quarter.1
Quarter.2
Quarter.3
Quarter.4

B. Use the DIR command to output a list of the Quarter files to the printer. Hand the results in to your instructor.

UNIT 5

ORGANIZATIONAL COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. use the MD command to make a subdirectory on a disk;
2. use the CD command to change subdirectories on a disk;
3. use the RD command to remove a subdirectory from a disk;
4. use the TREE command to view the subdirectories and files on a disk;
5. use the PATH command to direct information to and from subdirectories.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. CD
2. MD
3. PATH
4. RD
5. TREE.

ASSIGNMENTS

The assignments to be completed for this unit are

_____Guided Activity: Start the Computer and Make a Test File.
_____Guided Activity: Make Subdirectories on a Diskette.
_____Guided Activity: Change Subdirectories on a Diskette.
_____Guided Activity: Make Another Subdirectory.
_____Guided Activity: Copy a File to a Subdirectory.
_____Guided Activity: Name the Path.
_____Guided Activity: Set a New Path.
_____Guided Activity: View Subdirectories and Files with the TREE Command.
_____Guided Activity: Remove Subdirectories from a Diskette.
_____Review Questions.
_____Unit Exercises.
_____Documentation Research.

Place a check in front of those assigned by your instructor.

HARD-DISK MANAGEMENT COMMANDS

Imagine what it would be like if you only had one directory on a disk in which to store your files and programs. After a while the disk would become very slow and confusing to use. You would have a hard time trying to find those files and programs. DOS has solved this problem by giving you the ability to create subdirectories.

The root directory is the beginning directory on every disk; it is identified by the back-slash following the disk drive name. Subdirectories are created below the root directory. A directory can have files, labels, and subdirectories. The root directory of a double-sided diskette can hold 112 entries, whereas the root directory of a hard disk can hold 512 entries. Subdirectories are not restricted in the number of entries they can contain.

The hard-disk management commands MKDIR or MD (Make Directory), CHDIR or CD (Change Directory), RMDIR or RD (Remove Directory), TREE, and PATH give you the opportunity to effectively manage your subdirectories and the DOS system.

GUIDED ACTIVITY

START THE COMPUTER AND MAKE A TEXT FILE:

1. Turn on the computer.

2. Enter the current date and time.

NOTE: The commands in this unit are generally associated with use of a hard disk, but to avoid changing or disrupting lab equipment the guided activities will be practiced on a diskette.

3. Place a formatted disk in drive A:.
4. Type **A:**
5. Press <ENTER>.
6. Type **COPY CON TEST.TXT**
7. Press <ENTER>.
8. Type **This is a test.**
9. Press <ENTER>.
10. Press <F6>.
11. Press <ENTER>.
12. Type **CLS**
13. Press <ENTER>.

MD COMMAND

The MD or MKDIR (Make Subdirectory) command gives you the option of creating subdirectories. All subdirectories start from the root directory and move downward. In a filing cabinet the top drawer would be the root directory and the drawers below the top drawer would be subdirectories. You have only one root directory on a hard disk, and the root directory can hold 512 entries, but the subdirectories under the root directory can have as many entries as the amount of disk space will allow.

Subdirectories follow the same conventions as files. You can use up to eight characters in a subdirectory name. In the subdirectory name \WORD the back-slash is part of the subdirectory name. In the following exercise, we are going to create subdirectories for word processing (WORD), spreadsheet (SPD), data base (DATA), utility (UTIL).

GUIDED ACTIVITY

MAKE SUBDIRECTORIES ON A DISKETTE:

1. Type **MD\WORD**
2. Press <ENTER>.
3. Type **MD\SPD**
4. Press <ENTER>.
5. Type **MD\DATA**
6. Press <ENTER>.
7. Type **MD\UTIL**
8. Press <ENTER>.
9. Type **DIR**
10. Press <ENTER>.

Figure 5-1 shows the results of the MD command. Notice that in the expression "Directory of A:\" the back-slash is the name and identification of the root directory. The subdirectories WORD, SPD, DATA, and UTIL have the extension <DIR> which tells you they are subdirectories.

```
A:\>MD\WORD

A:\>MD\SPD

A:\>MD\DATA

A:\>MD\UTIL

A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

TEST       TXT         17   5-26-89   12:27a
WORD            <DIR>       5-26-89   12:28a
SPD             <DIR>       5-26-89   12:28a
DATA            <DIR>       5-26-89   12:28a
UTIL            <DIR>       5-26-89   12:28a
        5 File(s)    357376 bytes free

A:\>
```

Fig. 5-1. Making subdirectories.

11. Type **CLS**
12. Press <ENTER>.

CD COMMAND

The CD or CHDIR (Change Directory) command allows you to change from the root directory to the subdirectory you specify in the command.

GUIDED ACTIVITY

CHANGE SUBDIRECTORIES ON A DISKETTE: To change from the root directory to the \UTIL subdirectory, you would type **CD\UTIL** and press <ENTER>.

1. Type **CD\UTIL**
2. Press <ENTER>.
3. Type **DIR** to view the \UTIL subdirectory.

4. Press <ENTER>.

Figure 5-2 shows the screen contents of the \UTIL subdirectory and the response for the DIR command. Notice that the subdirectory is now called **A:\UTIL**. The single dot and double dot within the directory tell you that you are not in the root directory. The single dot refers to the current directory, and the double dot refers to the directory that is directly above the current directory. In the example, it is the root directory that is directly above the current directory. If you typed **DIR ..** and pressed <ENTER>, you would see the files in the root directory.

5. Type **CLS**

6. Press <ENTER>.

```
A:\>CD\UTIL

A:\UTIL>DIR

 Volume in drive A has no label
 Directory of  A:\UTIL

.            <DIR>          5-26-89  12:28a
..           <DIR>          5-26-89  12:28a
       2 File(s)    357376 bytes free

A:\UTIL>
```

Fig. 5-2. \UTIL subdirectory.

GUIDED ACTIVITY

MAKE ANOTHER SUBDIRECTORY: Another concept you should understand when dealing with subdirectories is the concept of path names. When you work with a file or subdirectory, you must tell DOS where it is located. DOS uses a path name to find its files and subdirectories.

In this activity you are going to add a subdirectory below the \UTIL subdirectory. This new directory will be called DOS. The path name for the DOS subdirectory file is \UTIL\DOS. Whenever you refer to this subdirectory, you must use the full path name of \UTIL\DOS. If you wanted to view the files on this subdirectory, for example, you would use the command syntax DIR\UTIL\DOS.

1. Type **MD\UTIL\DOS**
2. Press <ENTER>.
3. Type **DIR\UTIL\DOS**
4. Press <ENTER>.

Figure 5-3 shows the screen response to your command. Notice that the subdirectory is now called "A:\UTIL\DOS."

```
A:\UTIL>MD\UTIL\DOS

A:\UTIL>DIR\UTIL\DOS

 Volume in drive A has no label
 Directory of  A:\UTIL\DOS

.            <DIR>      5-26-89  12:31a
..           <DIR>      5-26-89  12:31a
       2 File(s)    356352 bytes free

A:\UTIL>
```

Fig. 5-3. UTIL\DOS subdirectory.

GUIDED ACTIVITY

COPY A FILE TO A SUBDIRECTORY: In this activity you are going to copy a file to the subdirectory \UTIL\DOS. To copy files to subdirectories, you use the

subdirectory's name with the command. To copy the file TEST.TXT from the root subdirectory of A:\ to the A:\UTIL\DOS subdirectory, you would use the following command syntax: `COPY A:\TEST.TXT A:\UTIL\DOS`.

1. Type **COPY A:\TEST.TXT A:\UTIL\DOS**
2. Press <ENTER>.
3. Type **DIR\UTIL\DOS** to view the subdirectory \UTIL\DOS and the file TEST.TXT.
4. Press <ENTER>.

Figure 5-4 shows the screen response to the command to copy the file TEST.TXT to the subdirectory \UTIL\DOS.

```
A:\UTIL>MD\UTIL\DOS

A:\UTIL>DIR\UTIL\DOS

 Volume in drive A has no label
 Directory of  A:\UTIL\DOS

.            <DIR>      5-26-89  12:31a
..           <DIR>      5-26-89  12:31a
       2 File(s)    356352 bytes free

A:\UTIL>COPY A:\TEST.TXT A:\UTIL\DOS
       1 File(s) copied

A:\UTIL>DIR\UTIL\DOS

 Volume in drive A has no label
 Directory of  A:\UTIL\DOS

.            <DIR>      5-26-89  12:31a
..           <DIR>      5-26-89  12:31a
TEST     TXT        17   5-26-89  12:27a
       3 File(s)    355328 bytes free

A:\UTIL>
```

Fig. 5-4. Copying a file to the A:\UTIL\DOS subdirectory

5. Type **CLS**
6. Press <ENTER>.

PATH COMMAND

The PATH command is an internal DOS command that is available after you have loaded DOS into your computer. The PATH command is used by DOS to enable it to find commands, as well as certain files that do not reside in the current default drive and subdirectory. When you initiate a command, DOS first looks at the current drive and subdirectory for the file or command. If it does not find it, you receive the error message "Bad command or file name." If you place the correct path name before the command or file, DOS will use that path to find and execute the command. DOS will execute a file or command with the extension of COM, EXE, or BAT. For example, in the subdirectory C:\DOS you have the file CHKDSK.COM. If you want to execute the CHKDSK.COM file from the root directory, you must use the command syntax `\DOS\CHKDSK` because the file does not reside in the root directory. The command tells DOS to go to the subdirectory `\DOS` and execute CHKDSK.COM.

One example of the use of the PATH command is to place the command in an AUTOEXEC.BAT file. If an AUTOEXEC.BAT file is present in the root directory of the computer's startup disk, the file will be read by DOS when the computer is started. This will allow the path to be loaded by DOS into the memory of the computer every time you start your computer. This will enable you to execute commands from any subdirectory listed by the PATH command. Use the following command syntax to direct DOS to search the subdirectories on drives C: and A:.

```
PATH=C:\DOS;A:\;A:\WORD;A:\SPD;A:\DATA;A:\UTIL;A:\UTIL\
DOS;
```

Remember that the DOS PATH command will only execute files with the EXE, COM, and BAT extensions.

GUIDED ACTIVITY

NAME THE PATH:

1. Type **`PATH;`** to set "No Path."
2. Press <ENTER>.
3. To display the current path, type **`PATH`**
4. Press <ENTER>.

Figure 5-5 will verify that there is no path set.

```
A:\UTIL>PATH;

A:\UTIL>PATH
No Path

A:\UTIL>
```

Fig. 5-5. PATH command.

When you issue the CHKDSK command and press <ENTER>, DOS does not find the file CHKDSK.COM and gives you a "Bad command or file name" message.

5. Type **CHKDSK**

6. Press <ENTER>.

Figure 5-6 shows that the screen response to the command will be "Bad command or file name" because DOS cannot find the CHKDSK.COM file in the current directory.

```
A:\UTIL>PATH;

A:\UTIL>PATH
No Path

A:\UTIL>CHKDSK
Bad command or file name

A:\UTIL>
```

Fig. 5-6. CHKDSK.COM not in the current path.

7. Type **C:\DOS\CHKDSK**

8. Press <ENTER>.

Figure 5-7 shows the screen response to the C:\DOS\CHKDSK command. This time the command was executed by DOS because a path to the command was available to DOS.

```
A:\UTIL>PATH;

A:\UTIL>PATH
No Path

A:\UTIL>CHKDSK
Bad command or file name

A:\UTIL>C:\DOS\CHKDSK

   362496 bytes total disk space
     5120 bytes in 5 directories
     2048 bytes in 2 user files
   355328 bytes available on disk

   655360 bytes total memory
   434176 bytes free

A:\UTIL>
```

Fig. 5-7. C:\DOS\CHKDSK command.

9. Type **CLS**
10. Press <ENTER>.

GUIDED ACTIVITY

SET A NEW PATH: The PATH command will allow DOS to search any subdirectory or drive in its path name. In the following example, if DOS does not find the file CHKDSK.COM in the subdirectory \UTIL, it will search in the root directory of C: and in the subdirectory C:\DOS for the file.

1. Type **PATH C:\DOS**
2. Press <ENTER>.
3. Type **PATH**
4. Press <ENTER>.
5. Type **CHKDSK**

6. Press <ENTER>.

Figure 5-8 shows the screen response to the command.

```
A:\UTIL>PATH C:\DOS

A:\UTIL>PATH
PATH=C:\DOS

A:\UTIL>CHKDSK

   362496 bytes total disk space
     5120 bytes in 5 directories
     2048 bytes in 2 user files
   355328 bytes available on disk

   655360 bytes total memory
   434176 bytes free

A:\UTIL>
```

Fig. 5-8. CHKDSK command.

7. Type **CLS**

8. Press <ENTER>.

TREE COMMAND

The TREE command will give you the opportunity to see the whole structure of the subdirectories on your disk. If you use the /F parameter, you can also see the files in the subdirectories.

GUIDED ACTIVITY

VIEW SUBDIRECTORIES AND FILES WITH THE TREE COMMAND:

1. Type **TREE A:/F**

2. Press <ENTER>.

3. DOS will scroll through the directories and list the files for you.

Figure 5-9 shows the subdirectory A:\UTIL\DOS with the file TEST.TXT.

```
Sub-directories:  None

Files:            None


Path: \UTIL

Sub-directories:  DOS

Files:            None


Path: \UTIL\DOS

Sub-directories:  None

Files:            TEST     .TXT


A:\>
```

Fig. 5-9. Viewing the subdirectory \UTIL\DOS with the TREE/F command.

4. Type **CLS**

5. Press <ENTER>.

RD COMMAND

A time will come when you need to remove a subdirectory from your disk. The RD (Remove Directory) command will accomplish the task of removing the subdirectory, but not the files within the subdirectory. Before you can remove a subdirectory from your disk, you must first remove all the files from that subdirectory except the dot files within the subdirectory. When you are ready to use the RD command, you must use it from the directory above the one to be removed. You may not remove the current or root directory.

GUIDED ACTIVITY

REMOVE SUBDIRECTORIES FROM A DISKETTE:

1. Type **CD** to return to the root directory.
2. Press <ENTER>.
3. Type **DIR** to be sure you are in the root directory.
4. Press <ENTER>.

Figure 5-10 shows the contents of the root directory.

```
A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

TEST       TXT         17   5-26-89   12:27a
WORD            <DIR>       5-26-89   12:28a
SPD             <DIR>       5-26-89   12:28a
DATA            <DIR>       5-26-89   12:28a
UTIL            <DIR>       5-26-89   12:28a
        5 File(s)    355328 bytes free

A:\>
```

Fig. 5-10. Root directory.

5. Type **DEL A:\UTIL\DOS\TEST.TXT** to delete the TEST.TXT file from the \UTIL\DOS subdirectory.
6. Press <ENTER>.
7. Type **DIR\UTIL\DOS**

8. Press <ENTER>.

Figure 5-11 shows that the file TEST.TXT has been deleted from the subdirectory. All that remains are the DOT files.

```
A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

TEST       TXT        17   5-26-89   12:27a
WORD            <DIR>      5-26-89   12:28a
SPD             <DIR>      5-26-89   12:28a
DATA            <DIR>      5-26-89   12:28a
UTIL            <DIR>      5-26-89   12:28a
       5 File(s)    355328 bytes free

A:\>DEL A:\UTIL\DOS\TEST.TXT

A:\>DIR\UTIL\DOS

 Volume in drive A has no label
 Directory of  A:\UTIL\DOS

.               <DIR>      5-26-89   12:31a
..              <DIR>      5-26-89   12:31a
       2 File(s)    356352 bytes free

A:\>
```

Fig. 5-11. Deleted TEST.TXT file.

9. Type **RD\UTIL\DOS**

10. Press <ENTER>.

11. Type **DIR\UTIL**

12. Press <ENTER>.

13. Type **RD\UTIL**

14. Press <ENTER>.

15. Type **RD\DATA**

16. Press <ENTER>.

17. Type **RD\SPD**
18. Press <ENTER>.
19. Type **RD\WORD**
20. Press <ENTER>.
21. Type **DEL A:TEST.TXT**
22. Press <ENTER>.
23. Type **DIR**
24. Press <ENTER>.

Figure 5-12 shows that all the subdirectories and the file TEST.TXT have been removed from the drive A:.

```
Volume in drive A has no label
Directory of  A:\UTIL

.            <DIR>      5-26-89  12:28a
..           <DIR>      5-26-89  12:28a
       2 File(s)    357376 bytes free

A:\>RD\UTIL

A:\>RD\DATA

A:\>RD\SPD

A:\>RD\WORD

A:\>DEL A:TEST.TXT

A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

File not found

A:\>
```

Fig. 5-12. Root directory.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`CD\DOS`	Changes from the root directory to the \DOS subdirectory.
`MD\DOS`	Makes a subdirectory called DOS.
`PATH C:\DOS;`	Directs DOS to look in both the root directory and the \DOS subdirectory for files.
`RD\DOS`	Will remove the \DOS subdirectory.
`TREE/F`	Allows you to see the subdirectories and files on your disks.

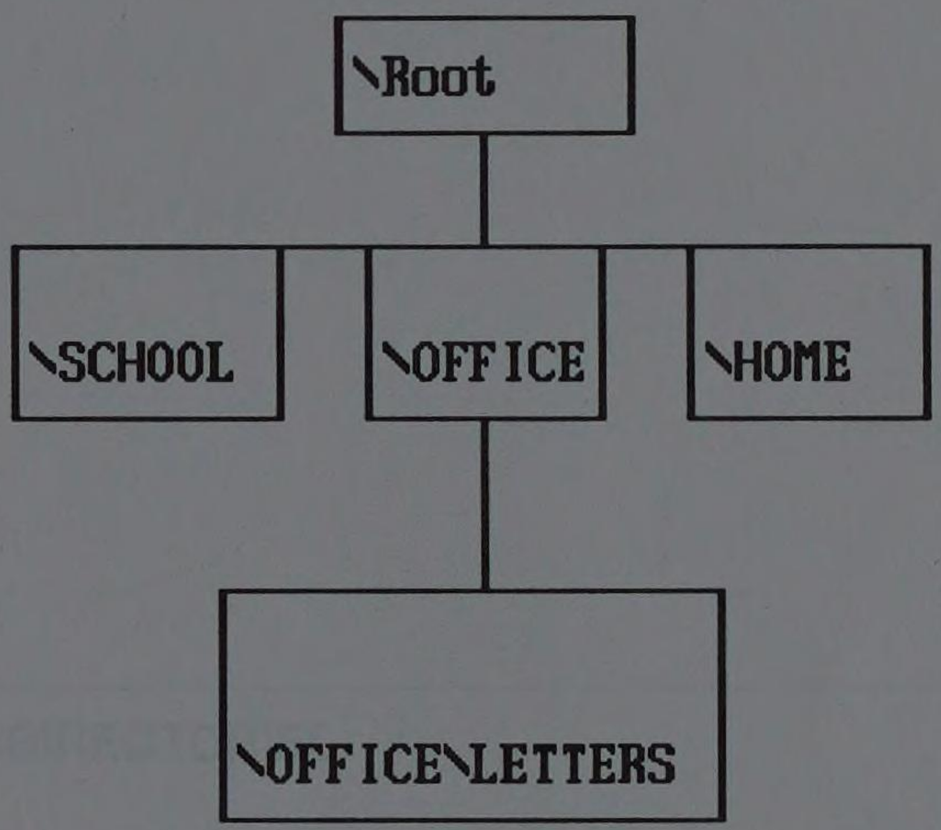
REVIEW QUESTIONS

1. How would you execute an internal DOS command on a subdirectory in a different drive?

2. Where would you place the PATH command so you would be able to initiate commands from a subdirectory that you are not currently using?

UNIT EXERCISES

1. Place a formatted disk in drive A:.
2. Make the subdirectories show in the diagram below.



3. Use the TREE command to output the subdirectory listing to your printer.
4. Set a path for the \OFFICE subdirectory.
5. Use the PATH command to show the path on the screen. Screen dump the results to your printer.
6. Remove all subdirectories from the disk in drive A:.

DOCUMENTATION RESEARCH

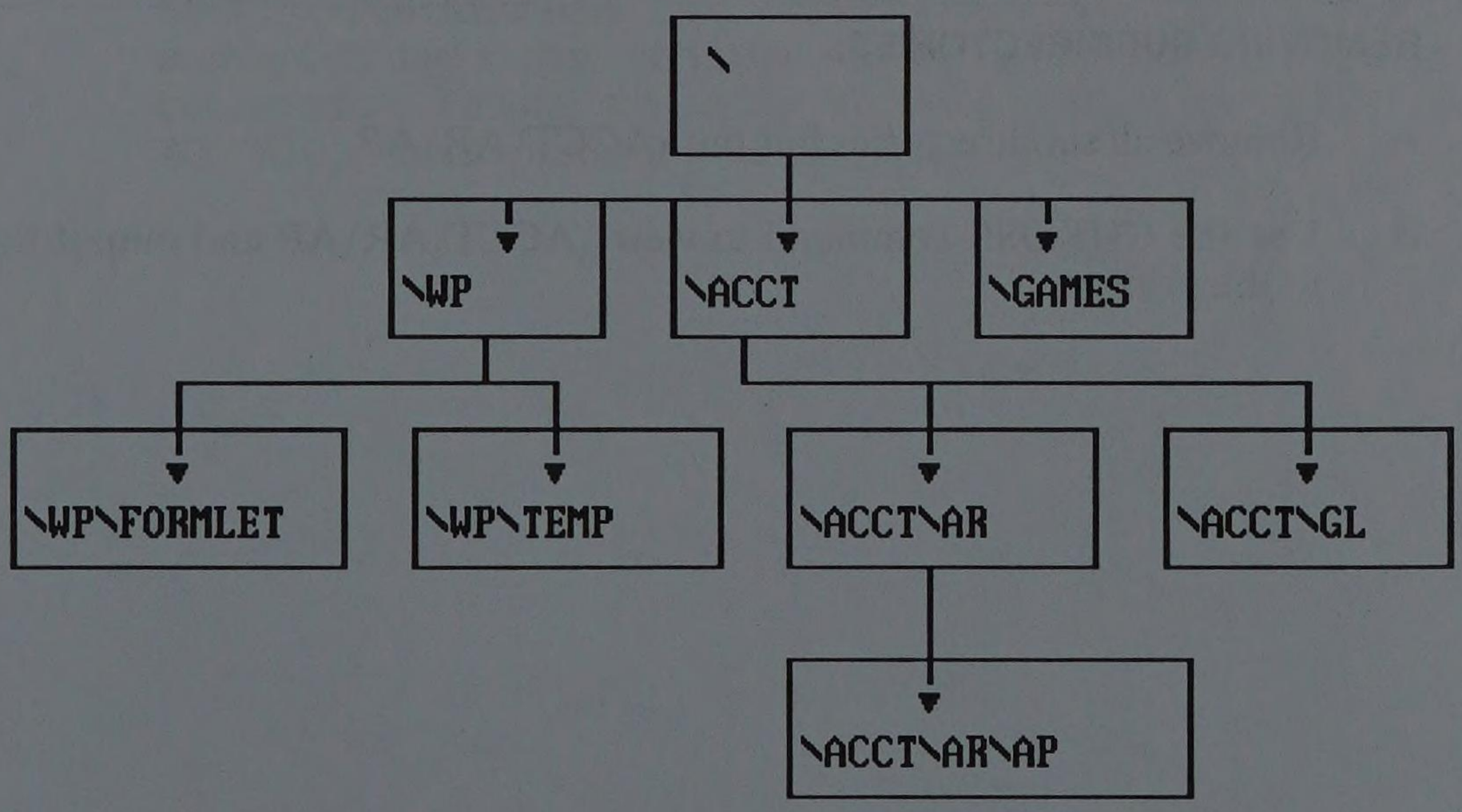
1. Research the *IBM Disk Operating System Version 3.3 Reference* manual for DOS commands and parameters not used in class.

APPLICATION C

PART 1

MAKING SUBDIRECTORIES:

A. Make the subdirectories shown in the diagram below.



PART 2

COPYING FILES TO SUBDIRECTORIES:

A. Copy the months of the year.

   1. Copy the file JAN.88 into \WP.
   2. Copy the file FEB.88 into \ACCT.
   3. Copy the file MAR.88 into \GAMES.
   4. Copy the files JAN.88, FEB.88, and MAR.88 into \WP\FORMLET.
   5. Copy the files APR.88, MAY.88, and JUN.88 into \WP\TEMP.
   6. Copy the files JUL.88, AUG.88, and SEPT.88 into \ACCT\AR.
   7. Copy the files OCT.88, NOV.88, and DEC.88 into \ACCT\GL.
   8. Copy all files into \ACCT\AR\AP.

B. Use the TREE/F command, and output the results of the command to the printer.

PART 3

REMOVING SUBDIRECTORIES:

A. Remove all subdirectories but the \ACCT\AR\AP.

B. Use the CHKDSK command to view \ACCT\AR\AP and output the results to the printer.

PART III

INTERMEDIATE DOS COMMANDS

The third part of this text covers DOS commands for the intermediate user. Each of the commands goes beyond the needs of the novice user who is mainly concerned with using application files, formatting disks, and storing data files.

In the following units you will experiment with the DOS line editor, batch files, and the safety commands.

EDLIN, the line editor, can be used to create batch files, such as the commonly used AUTOEXEC.BAT and CONFIG.SYS files. Batch files can be used to accomplish tasks that normally require keyboard execution of several DOS commands. Finally, the safety commands (like BACKUP, RESTORE, and ATTRIB) provide protection for your data.

UNIT 6

THE DOS EDITING COMMAND

OBJECTIVES

After completing this unit, you will be able to

1. write a text file;
2. edit a text file;
3. search and replace words within the file;
4. move text to other lines within the file;
5. type a text file to the screen.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. EDLIN.

ASSIGNMENTS

The assignments to be completed for this unit are

____Guided Activity: Start EDLIN.
____Guided Activity: Use the INSERT Command.
____Guided Activity: Insert a Line into an Existing File.
____Guided Activity: List Lines.
____Guided Activity: Use the DELETE Command.
____Guided Activity: End and Save an EDLIN Text File.
____Guided Activity: Output to the Screen with the TYPE Command.
____Guided Activity: Use the Q Command to Abort a File.
____Guided Activity: Create a New EDLIN File.
____Guided Activity: Make an Office Help Screen.
____Guided Activity: Insert New Lines.
____Guided Activity: Use the EDIT Command.
____Guided Activity: Use the COPY Command.
____Guided Activity: Use the MOVE Command
____Guided Activity: Use the SEARCH Command
____Guided Activity: Use the SEARCH Command with a ?.
____Guided Activity: Use the REPLACE Command.
____Guided Activity: Clean Up and Save the File.
____Review Questions.
____Unit Exercises.
____Documentation Research.

Place a check in front of those assigned by your instructor.

EDLIN: THE TEXT EDITOR

EDLIN is a text editor that allows you to write and edit text files. EDLIN is an external DOS command, so you will need the EDLIN.COM file in order to use the command. By typing **`EDLIN A:ADDRESS.TXT`** at the C:\DOS> system prompt, you begin a new EDLIN file named ADDRESS.TXT, and the file will be written to drive A: when it is stored.

GUIDED ACTIVITY

STARTING EDLIN: If you want to save the EDLIN file on a drive other than the default drive you must identify the drive in the EDLIN command. The EDLIN file does not have to be on the same disk as EDLIN.COM.

1. Type **`EDLIN A:ADDRESS.TXT`**
2. Press <ENTER>.

When you start a new file, EDLIN will state this at the beginning of the file.

Figure 6-1 shows the new EDLIN file ADDRESS.TXT.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*
```

Fig. 6-1. New EDLIN file ADDRESS.TXT.

GUIDED ACTIVITY

USE THE INSERT COMMAND: When you start a new EDLIN file, you must use the i, INSERT LINE command. The first line of the text file must be inserted using the INSERT command. EDLIN automatically gives each line of text a line number when you press <ENTER>. Later, you will see that these line numbers make it easier to edit the EDLIN file.

The asterisk is the system prompt used by EDLIN. All EDLIN commands start at the prompt. This is the starting point for your new file ADDRESS.TXT. EDLIN will allow you to type 253 characters on a line, but it will not hyphenate the words when it moves to the next line.

1. Type `i`
2. Press <ENTER>.

Figure 6-2 shows the screen response to the INSERT command.

3. Type **Tyler Euthon**

4. Press <ENTER>.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*
1:*
```

Fig. 6-2. INSERT command.

You are now ready to enter lines 2 and 3. Remember, when you start typing, the words will appear on the numbered line 2.

5. Type **147 Houston Street**

6. Press <ENTER>.

7. Type **Mineola, NY 11501**

8. Press <ENTER>.

Figure 6-3 shows the inserted lines.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*Tyler Euthon
       2:*147 Houston Street
       3:*Mineola, NY 11501
       4:*
```

Fig. 6-3. Inserted lines.

GUIDED ACTIVITY

INSERT A LINE INTO AN EXISTING FILE: EDLIN does not give you the ability to just move up a line and start typing a new line. You must first leave the present insert operation and start the insert function again. Pressing <CTRL-C> stops the insert operation.

1. Press <CTRL-C> to stop inserting lines.

Figure 6-4 shows the screen response where you stop the insert operation.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*Tyler Euthon
       2:*147 Houston Street
       3:*Mineola, NY 11501
       4:*^C

*
*
```

Fig. 6-4. Stopping the insert operation.

Now you can insert new lines into your file. When you insert new lines into the existing file, all lines will be moved forward and renumbered automatically, although you will not see the renumbered lines until you use the LIST command to list the lines in the file.

2. Type **2i**
3. Press <ENTER>.
4. Type **President**
5. Press <ENTER>.
6. Type **Hot Air Balloon Company**
7. Press <ENTER>.
8. Press <CTRL-C> to stop the insert function.

Figure 6-5 shows the screen response when you insert new lines and stop the insert function.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*Tyler Euthon
       2:*147 Houston Street
       3:*Mineola, NY  11501
       4:*^C

*2i
       2:*President
       3:*Hot Air Balloon Company
       4:*^C

*
```

Fig. 6-5. Inserting and stopping the insert function.

GUIDED ACTIVITY

LIST LINES: The LIST command is used to display lines within your file. You have the ability to list a single line or a group of lines. The command is very straightforward. To list line 1, you type **1,1L** and press <ENTER>. If you type the command **1L** and press <ENTER>, you will list 23 lines, starting with line 1. To list a group of lines between lines 1 and 3, you type **1,3L** and press <ENTER>.

1. Type **1L**
2. Press <ENTER>.

Figure 6-6 shows the screen response to the LIST command.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*Tyler Euthon
       2:*147 Houston Street
       3:*Mineola, NY  11501
       4:*^C

*2i
       2:*President
       3:*Hot Air Balloon Company
       4:*^C

*1L
       1: Tyler Euthon
       2: President
       3: Hot Air Balloon Company
       4:*147 Houston Street
       5: Mineola, NY  11501
*
```

Fig. 6-6. LIST command.

If you wanted to list only lines 2 through 4, you would enter the command `2,4L`.

3. Type **2,4L**
4. Press <ENTER>.

Figure 6-7 shows the screen response to the `2-4L` command.

```
C:\DOS>EDLIN A:ADDRESS.TXT
New file
*i
       1:*Tyler Euthon
       2:*147 Houston Street
       3:*Mineola, NY  11501
       4:*^C

*2i
       2:*President
       3:*Hot Air Balloon Company
       4:*^C

*1L
       1: Tyler Euthon
       2: President
       3: Hot Air Balloon Company
       4:*147 Houston Street
       5: Mineola, NY  11501
*2,4L
       2: President
       3: Hot Air Balloon Company
       4:*147 Houston Street
*
```

Fig. 6-7. LIST command.

GUIDED ACTIVITY

USE THE DELETE COMMAND: To delete a line of text in EDLIN, you only have to type the line number followed by the letter "D." To delete line 1 of the current file, type **1D** and press <ENTER>. To delete lines 2 and 3 of the current file, type **2,3D** and press <ENTER>. To delete lines 1 through 3, type **1,3D** and press <ENTER>.

1. Type **2,3D**
2. Press **<ENTER>**.
3. To list the first 23 lines of the file, type **L**
4. Press **<ENTER>**.

Figure 6-8 shows the response to the 2,3D command.

```
        2:*147 Houston Street
        3:*Mineola, NY  11501
        4:*^C

*2i
        2:*President
        3:*Hot Air Balloon Company
        4:*^C

*1L
        1: Tyler Euthon
        2: President
        3: Hot Air Balloon Company
        4:*147 Houston Street
        5: Mineola, NY  11501
*2,4L
        2: President
        3: Hot Air Balloon Company
        4:*147 Houston Street
*2,3D
*L
        1: Tyler Euthon
        2:*147 Houston Street
        3: Mineola, NY  11501
*
```

Fig. 6-8. DELETE command.

GUIDED ACTIVITY

END AND SAVE AN EDLIN TEXT FILE: To end EDLIN and save your file, you must type the letter **E** at the asterisk prompt and press <ENTER>.

1. Type **E**
2. Press <ENTER>.

Figure 6-9 shows the screen response when you end and save an EDLIN file.

```
        4:*^C

*2i
        2:*President
        3:*Hot Air Balloon Company
        4:*^C

*1L
        1: Tyler Euthon
        2: President
        3: Hot Air Balloon Company
        4:*147 Houston Street
        5: Mineola, NY  11501
*2,4L
        2: President
        3: Hot Air Balloon Company
        4:*147 Houston Street
*2,3D
*L
        1: Tyler Euthon
        2:*147 Houston Street
        3: Mineola, NY  11501
*E

C:\DOS>
```

Fig. 6-9. Ending and saving the ADDRESS.TXT file.

GUIDED ACTIVITY

OUTPUT TO THE SCREEN WITH THE TYPE COMMAND:

1. Type **CLS**
2. Press <ENTER>.
3. Type **TYPE A:ADDRESS.TXT**
4. Press <ENTER>.

Figure 6-10 shows the screen response to the TYPE command.

5. Type **CLS**
6. Press <ENTER>.

```
C:\DOS>TYPE A:ADDRESS.TXT
Tyler Euthon
147 Houston Street
Mineola, NY  11501

C:\DOS>
```

Fig. 6-10. TYPE A:ADDRESS.TXT command.

GUIDED ACTIVITY

USE THE Q COMMAND TO ABORT A FILE: An EDLIN file can be aborted any time the cursor is at the EDLIN prompt. If you type **EDLIN A:SCREEN.TXX** and press <ENTER> and then decide that this is not the filename you want to use, just type **Q**. Then type **Y** to EDLIN's request to abort the file.

1. Type **EDLIN A:SCREEN.TXX**
2. Press **<ENTER>**.
3. Type **Q**
4. Press **<ENTER>**.
5. Type **Y**

Figure 6-11 shows the aborting of a new file.

```
C:\DOS>EDLIN A:SCREEN.TXX
New file
*Q
Abort edit (Y/N)? Y
C:\DOS>
```

Fig. 6-11. Aborting the SCREEN.TXX file.

6. Type **CLS**
7. Press <ENTER>.

GUIDED ACTIVITY

CREATE A NEW EDLIN FILE:

1. Type **EDLIN A:SCREEN.TXT**
2. Press <ENTER>.
3. Type **i**
4. Press <ENTER>.
5. Type **It has been a good day.**
6. Press <ENTER>.

7. Type **Pack up your computer.**
8. Press <ENTER>.
9. Type **Good night.**
10. Press <ENTER>.
11. Type **Bye.**
12. Press <ENTER>.
13. Press <CTRL-C>.
14. Type **L**
15. Press <ENTER>.
16. Type **1D**
17. Press <ENTER>.
18. Type **L**
19. Press <ENTER>.
20. Type **1,3D**
21. Press <ENTER>.
22. Type **L**
23. Press <ENTER>.
24. Type **Q**
25. Press <ENTER>.
26. Type **Y**

MORE EDLIN

In this section you will create more EDLIN files and use the COPY, SEARCH, and REPLACE commands.

GUIDED ACTIVITY

MAKE AN OFFICE HELP SCREEN:

1. Type **EDLIN A:SCREEN.TXT**
2. Press <ENTER>.
3. Type **i**
4. Press <ENTER>.
5. Type **B & C OFFICE HELP SCREEN.**
6. Press <ENTER>.
7. To create a blank line, press <ENTER>.
8. Type **Press number 1 to start your Wordprocessing Program.**
9. Press <ENTER>.
10. Press <CTRL-C>.
11. Type **1i**
12. Press <ENTER>.
13. Type **GOOD MORNING.**
14. Press <ENTER>.
15. Press <ENTER>.
16. Press <CTRL-C>.
17. Type **1L**
18. Press <ENTER>.

Figure 6-12 shows the screen response to the new EDLIN file SCREEN.TXT.

```
C:\DOS>EDLIN A:SCREEN.TXT
New file
*i
       1:*B & C OFFICE HELP SCREEN.
       2:*
       3:*Press number 1 to start your Wordprocessing Program.
       4:*^C

*1i
       1:*GOOD MORNING.
       2:*
       3:*^C

*1L
       1: GOOD MORNING.
       2:
       3:*B & C OFFICE HELP SCREEN.
       4:
       5: Press number 1 to start your Wordprocessing Program.
*
```

Fig. 6-12. New SCREEN.TXT file.

GUIDED ACTIVITY

INSERT NEW LINES: If you will notice, line 3 has an asterisk at the beginning of the line. It means that this line is the active line. If you press <ENTER> now, line 4 will appear on the screen, and you will be able to add text to that line. If you want to add text to the last line of your file, use the command #i. This will give you the next line number after your last line to add text.

1. Type **#i**
2. Press <ENTER>.
3. Type **Don't forget to adjust your monitor.**
4. Press <ENTER>.
5. Type **Place your data disk in drive A:.**
6. Press <ENTER>.

Figure 6-13 shows the screen response to the #i command.

```
C:\DOS>EDLIN A:SCREEN.TXT
New file
*i
       1:*B & C OFFICE HELP SCREEN.
       2:*
       3:*Press number 1 to start your Wordprocessing Program.
       4:*^C

*1i
       1:*GOOD MORNING.
       2:*
       3:*^C

*1L
       1: GOOD MORNING.
       2:
       3:*B & C OFFICE HELP SCREEN.
       4:
       5: Press number 1 to start your Wordprocessing Program.
*#i
       6:*
```

Fig. 6-13. #i command.

7. Press <CTRL-C>.

You are now going to add a text line in the middle of your file. When you do this at line 6, all lines of text will be moved forward and renumbered.

8. Type **6i**

9. Press <ENTER>.

10. Type **Make sure you will select the correct program number.**

11. Press <ENTER>.

12. Type **Back up your files when you are done.**

13. Press <ENTER>.

14. Press <CTRL-C>.

Figure 6-14 shows the screen response to the line number insertion at line 6.

```
        3:*Press number 1 to start your Wordprocessing Program.
        4:*^C

*1i
        1:*GOOD MORNING.
        2:*
        3:*^C

*1L
        1: GOOD MORNING.
        2:
        3:*B & C OFFICE HELP SCREEN.
        4:
        5: Press number 1 to start your Wordprocessing Program.
*#i
        6:*Don't forget to adjust your monitor.
        7:*Place your data disk in drive A:.
        8:*^C

*6i
        6:*Make sure you will select the correct program number.
        7:*Back up your files when you are done.
        8:*^C

*
```

Fig. 6-14. Line number insertion.

GUIDED ACTIVITY

USE THE EDIT COMMAND: Now you will correct and change text at line 6. You should be at the EDLIN asterisk prompt.

1. Type **6**
2. Press **<ENTER>**.

There are two methods of copying characters across a line. The first method is to press the <RIGHT ARROW> Key. As you do, the characters on the existing line will be copied to the new line. The other method is to press the <F2> Key and a character at the point where you want to move. For example, if you want to move to the character "w" in line 6, you would press <F2> and then w, and the cursor would move to the first occurrence of a "w" in the line.

3. Press **<RIGHT ARROW>** until your cursor is under the letter "w" in the word "will".

4. Press <LEFT ARROW> until you reach the beginning of the line.
5. Press <F2>.
6. Type **w**
7. Type **select the correct program number.**
8. Press <ENTER>.
9. Type **6**
10. Press <ENTER>.
11. Press <F2>.
12. Type **p**
13. Press <INS>. Pressing the Insert Key will allows you to insert text into the line.
14. Type **Wordprocessing**
15. Press <SPACE BAR> one time.
16. Press <F3>. This will copy the remainder of the text in the line.
17. Press <ENTER>.
18. Type **1L**
19. Press <ENTER>.

Figure 6-15 shows the screen results of editing the file.

```
        7:*Place your data disk in drive A:.
        8:*^C

*6i
        6:*Make sure you will select the correct program number.
        7:*Back up your files when you are done.
        8:*^C

*6
        6:*Make sure you will select the correct program number.
        6:*Make sure you select the correct program number.
*6
        6:*Make sure you select the correct program number.
        6:*Make sure you select the correct Wordprocessing program number.
*1L
        1: GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Press number 1 to start your Wordprocessing Program.
        6:*Make sure you select the correct Wordprocessing program number.
        7: Back up your files when you are done.
        8: Don't forget to adjust your monitor.
        9: Place your data disk in drive A:.
*
```

Fig. 6-15. EDIT command.

GUIDED ACTIVITY

USE THE COPY COMMAND: You are going to copy a line in your file to another location. Type the beginning line number followed by a comma, the end line number followed by a comma, the number sign, and the letter "C", followed by a semicolon. To copy line 1 to the end of the file you would enter **`1,1,#C;`**.

1. Type **`1,1,#C;`** to copy "GOOD MORNING." in line 1 to line 10.
2. Press <ENTER>.
3. Type **`1L`**
4. Press <ENTER>.

Figure 6-16 shows the results of copying a single line with the COPY command.

```
      6:*Make sure you select the correct program number.
      6:*Make sure you select the correct Wordprocessing program number.
*1L
      1: GOOD MORNING.
      2:
      3: B & C OFFICE HELP SCREEN.
      4:
      5: Press number 1 to start your Wordprocessing Program.
      6:*Make sure you select the correct Wordprocessing program number.
      7: Back up your files when you are done.
      8: Don't forget to adjust your monitor.
      9: Place your data disk in drive A:.
*1,1,#C;
*1L
      1: GOOD MORNING.
      2:
      3: B & C OFFICE HELP SCREEN.
      4:
      5: Press number 1 to start your Wordprocessing Program.
      6: Make sure you select the correct Wordprocessing program number.
      7: Back up your files when you are done.
      8: Don't forget to adjust your monitor.
      9: Place your data disk in drive A:.
     10: GOOD MORNING.
*
```

Fig. 6-16. COPY command.

5. Type **2,3,11C;** to copy lines 2 and 3 to the end of the file.
6. Press <ENTER>.
7. Press <CTRL-C>.
8. Type **1L**
9. Press <ENTER>.

Figure 6-17 shows the screen results of more copying with the COPY command.

```
        4:
        5: Press number 1 to start your Wordprocessing Program.
        6: Make sure you select the correct Wordprocessing program number.
        7: Back up your files when you are done.
        8: Don't forget to adjust your monitor.
        9: Place your data disk in drive A:.
       10: GOOD MORNING.
*2,3,11C;
       12:*B & C OFFICE HELP SCREEN.
       12:*^C

*1L
        1: GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Press number 1 to start your Wordprocessing Program.
        6: Make sure you select the correct Wordprocessing program number.
        7: Back up your files when you are done.
        8: Don't forget to adjust your monitor.
        9: Place your data disk in drive A:.
       10: GOOD MORNING.
       11:
       12:*B & C OFFICE HELP SCREEN.
*
```

Fig. 6-17. More copying with the COPY command.

GUIDED ACTIVITY

USE THE MOVE COMMAND: The MOVE command can be used to move lines of text within your file. If you want to reverse line 8 with line 9 in your file, you would issue the command `8,8,10M;`. Do not leave any spaces in the command. The command will cause line 8 to move in front of line 10 and line 9 to drop into the place of line 8.

1. Type **`8,8,10M;`**
2. Press <ENTER>.
3. Press <CTRL-C>.
4. Type **`1L`**
5. Press <ENTER>.

Figure 6-18 shows the screen response after moving line 8 with the MOVE command.

```
       6: Make sure you select the correct Wordprocessing program number.
       7: Back up your files when you are done.
       8: Don't forget to adjust your monitor.
       9: Place your data disk in drive A:.
      10: GOOD MORNING.
      11:
      12:*B & C OFFICE HELP SCREEN.
*8,8,10M;
      10:*GOOD MORNING.
      10:*^C

*1L
       1: GOOD MORNING.
       2:
       3: B & C OFFICE HELP SCREEN.
       4:
       5: Press number 1 to start your Wordprocessing Program.
       6: Make sure you select the correct Wordprocessing program number.
       7: Back up your files when you are done.
       8: Place your data disk in drive A:.
       9: Don't forget to adjust your monitor.
      10:*GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*
```

Fig. 6-18. MOVE command.

6. Type **5,5,7M;** to reverse line 6 and line 5.
7. Press <ENTER>.
8. Press <CTRL-C>.
9. Type **1L**
10. Press <ENTER>.

Figure 6-19 shows the screen response when reversing line 5 and line 6.

```
       6: Make sure you select the correct Wordprocessing program number.
       7: Back up your files when you are done.
       8: Place your data disk in drive A:.
       9: Don't forget to adjust your monitor.
      10:*GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*5,5,7M;
       7:*Back up your files when you are done.
       7:*^C

*1L
       1: GOOD MORNING.
       2:
       3: B & C OFFICE HELP SCREEN.
       4:
       5: Make sure you select the correct Wordprocessing program number.
       6: Press number 1 to start your Wordprocessing Program.
       7:*Back up your files when you are done.
       8: Place your data disk in drive A:.
       9: Don't forget to adjust your monitor.
      10: GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*
```

Fig. 6-19. Reversing line 5 and line 6.

11. Type **7,7,10M;** to move line 7 to line 9 and move the old line 9 to line 8.
12. Press <ENTER>.
13. Press <CTRL-C>.
14. Type **1L**
15. Press <ENTER>.

Figure 6-20 shows the MOVE command moving line 7 to the position of line 9 and line 9 to the position of line 8.

```
       6: Press number 1 to start your Wordprocessing Program.
       7:*Back up your files when you are done.
       8: Place your data disk in drive A:.
       9: Don't forget to adjust your monitor.
      10: GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*7,7,10M;
      10:*GOOD MORNING.
      10:*^C

*1L
       1: GOOD MORNING.
       2:
       3: B & C OFFICE HELP SCREEN.
       4:
       5: Make sure you select the correct Wordprocessing program number.
       6: Press number 1 to start your Wordprocessing Program.
       7: Place your data disk in drive A:.
       8: Don't forget to adjust your monitor.
       9: Back up your files when you are done.
      10:*GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*
```

Fig. 6-20. Changing line 7 to line 9 with MOVE command.

GUIDED ACTIVITY

USE THE SEARCH COMMAND: It is often helpful to be able to find words or text within a file. EDLIN gives you this ability with the SEARCH command. In this activity, you will find the word "**GOOD**" with the SEARCH command. To find the word in your file, you list the first line number (1) followed by a comma, then the last line number (12) followed by S (which stands for search), and the word you want to find. It is important not to leave any spaces in the command. The command looks like this: `1,12SGOOD`

1. Type **`1,12SGOOD`**
2. Press <ENTER>.

Figure 6-21 shows the results of the SEARCH command.

```
        8: Place your data disk in drive A:.
        9: Don't forget to adjust your monitor.
       10: GOOD MORNING.
       11:
       12: B & C OFFICE HELP SCREEN.
*7,7,10M:
       10:*GOOD MORNING.
       10:*^C

*1L
        1: GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Make sure you select the correct Wordprocessing program number.
        6: Press number 1 to start your Wordprocessing Program.
        7: Place your data disk in drive A:.
        8: Don't forget to adjust your monitor.
        9: Back up your files when you are done.
       10:*GOOD MORNING.
       11:
       12: B & C OFFICE HELP SCREEN.
*1,12SGOOD
        1: GOOD MORNING.
*
```

Fig. 6-21. SEARCH command.

GUIDED ACTIVITY

USE THE SEARCH COMMAND WITH ?: In this activity, you will use the SEARCH command with a ?. Using a question mark in your string command causes EDLIN to search the file and, upon finding the word specified, to ask you if it is the word. If you say yes, the search ends. If you say no, the search will continue until the whole file has been searched.

1. Type **1,12?SGOOD**
2. Press **<ENTER>**.
3. Type **N**
4. Type **N**

Figure 6-22 shows the screen response to the `1,12S?GOOD` command. In line 1, EDLIN asks if the word found is O.K. If you type **N**, it will proceed to the next occurrence of the word and ask you again. When it has gone through the whole file from line 1 through line 12, it will stop.

```
      10:*GOOD MORNING.
      10:*^C

*1L
       1: GOOD MORNING.
       2:
       3: B & C OFFICE HELP SCREEN.
       4:
       5: Make sure you select the correct Wordprocessing program number.
       6: Press number 1 to start your Wordprocessing Program.
       7: Place your data disk in drive A:.
       8: Don't forget to adjust your monitor.
       9: Back up your files when you are done.
      10:*GOOD MORNING.
      11:
      12: B & C OFFICE HELP SCREEN.
*1,12SGOOD
       1: GOOD MORNING.
*1,12?SGOOD
       1:*GOOD MORNING.
O.K.? N
      10: GOOD MORNING.
O.K.? N
Not found
*
```

Fig. 6-22. SEARCH command with the ?.

5. Type **1,12?SGOOD**
6. Press <ENTER>.
7. Type **Y**
8. Type **1L**
9. Press <ENTER>.

Figure 6-23 shows the SEARCH command used to find the word "GOOD".

```
*1,12SGOOD
        1: GOOD MORNING.
*1,12?SGOOD
        1:*GOOD MORNING.
O.K.? N
       10: GOOD MORNING.
O.K.? N
Not found
*1,12?SGOOD
        1:*GOOD MORNING.
O.K.? Y
*1L
        1:*GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Make sure you select the correct Wordprocessing program number.
        6: Press number 1 to start your Wordprocessing Program.
        7: Place your data disk in drive A:.
        8: Don't forget to adjust your monitor.
        9: Back up your files when you are done.
       10: GOOD MORNING.
       11:
       12: B & C OFFICE HELP SCREEN.
*
```

Fig. 6-23. SEARCH command used to find a word.

GUIDED ACTIVITY

USE THE REPLACE COMMAND: You can use the REPLACE command to find "MORNING", and replace "MORNING" with the word "NIGHT". The command structure for the REPLACE command is similar to the SEARCH command. You must give the beginning line number, a comma, then the end line number, followed by a question mark, an R for the REPLACE command, and then the word you are looking for. You must separate the words you are looking for and the word you want to replace with ^Z. The characters ^Z are made by pressing the <CTRL-Z>. The command looks like this: `1,12?RMORNING.^ZNIGHT.`.

1. Type **`1,12?RMORNING.^ZNIGHT.`**
2. Press **<ENTER>**.
3. Type **N**
4. Type **Y**

Figure 6-24 shows the screen response to the REPLACE command using the string to find and replace the word "MORNING" with the word "NIGHT".

```
      10: GOOD MORNING.
O.K.? N
Not found
*1,12?SGOOD
       1:*GOOD MORNING.
O.K.? Y
*1L
       1:*GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Make sure you select the correct Wordprocessing program number.
        6: Press number 1 to start your Wordprocessing Program.
        7: Place your data disk in drive A:.
        8: Don't forget to adjust your monitor.
        9: Back up your files when you are done.
       10: GOOD MORNING.
       11:
       12: B & C OFFICE HELP SCREEN.
*1,12?RMORNING.^ZNIGHT.
       1:*GOOD NIGHT.
O.K.? N
      10: GOOD NIGHT.
O.K.? Y
*
```

Fig. 6-24. REPLACE command.

5. Type **1L**
6. Press <ENTER>.

Figure 6-25 shows the screen results of the REPLACE command.

```
        7: Place your data disk in drive A:.
        8: Don't forget to adjust your monitor.
        9: Back up your files when you are done.
       10: GOOD MORNING.
       11:
       12: B & C OFFICE HELP SCREEN.
*1,12?RMORNING.^ZNIGHT.
        1:*GOOD NIGHT.
O.K.? N
       10: GOOD NIGHT.
O.K.? Y
*1L
        1: GOOD MORNING.
        2:
        3: B & C OFFICE HELP SCREEN.
        4:
        5: Make sure you select the correct Wordprocessing program number.
        6: Press number 1 to start your Wordprocessing Program.
        7: Place your data disk in drive A:.
        8: Don't forget to adjust your monitor.
        9: Back up your files when you are done.
       10:*GOOD NIGHT.
       11:
       12: B & C OFFICE HELP SCREEN.
*
```

Fig. 6-25. Results of the REPLACE command.

GUIDED ACTIVITY

CLEAN UP AND SAVE THE FILE:

1. Type **`11,12D`**
2. Press <ENTER>.
3. Type **`1L`**
4. Press <ENTER>.
5. Type **`E`**
6. Press <ENTER>.
7. Type **`CLS`**
8. Press <ENTER>.
9. Type **`TYPE A:SCREEN.TXT`** to view your finished file.

10. Press <ENTER>.

Figure 6-26 shows the completed SCREEN.TXT file.

```
C:\DOS>TYPE A:SCREEN.TXT
GOOD MORNING.

B & C OFFICE HELP SCREEN.

Make sure you select the correct Wordprocessing program number.
Press number 1 to start your Wordprocessing Program.
Place your data disk in drive A:.
Don't forget to adjust your monitor.
Back up your files when you are done.
GOOD NIGHT.

C:\DOS>
```

Fig. 6-26. SCREEN.TXT file.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

CTRL-C	Stops editing lines.
EDLIN NEWFILE.TXT	Loads EDLIN and creates a new file.
EDLIN OLDFILE.TXT	Loads EDLIN and allows you to edit an old file.
I or i	Inserts a line into an EDLIN file.
L	Lists up to 23 lines in an EDLIN file.
Q	Aborts an EDLIN file.

REVIEW QUESTIONS

1. How do you quit EDLIN without saving the text file?
2. What will EDLIN do if you do not use a ? in the SEARCH and REPLACE command?
3. Do you have to save your EDLIN file on the same disk with EDLIN.COM?

UNIT EXERCISES

1. Use EDLIN to create a text file of each day of the week, and call the file WEEK.TXT.
2. Use the TYPE command to display the file on the screen. Make a screen dump of the file on the screen, and hand in the results to your instructor.
3. Use EDLIN to edit the WEEK.TXT file by placing the current calendar date for each day of the week in the file. Make a screen dump of the file before you save it in EDLIN.

DOCUMENTATION RESEARCH

1. Research the *IBM Disk Operating System Version 3.3 Reference* manual for DOS commands and parameters not used in class.

APPLICATION D

PART 1

FORMAT A BLANK DISKETTE: In this application, you will need a blank diskette to store your files.

A. Format a blank diskette as a system diskette.

PART 2

CREATING TEXT FILES USING EDLIN:

A. Create the seasons of the year as text files and add the appropriate months for each season.

PART 3

USING THE TYPE COMMAND:

A. Use the TYPE command to display the files on the screen. Screen dump the files when the screen has all the appropriate months. Give the results to your instructor.

UNIT 7

THE INVISIBLE DOS COMMAND

OBJECTIVES

After completing this unit, you will be able to

1. create a BATCH file;
2. create an AUTOEXEC.BAT file;
3. create a CONFIG.SYS file.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. BUFFERS
2. ECHO
3. FILES
4. PAUSE.

ASSIGNMENTS

The assignments to be completed for this unit are

____Guided Activity: Create the First BAT File.
____Guided Activity: Add invisible lines to the HELLO.BAT File.
____Guided Activity: Terminate a Batch File.
____Guided Activity: Stop the BAT File with the PAUSE Command.
____Guided Activity: Create an AUTOEXEC.BAT File.
____Guided Activity: Reboot the Computer with AUTOEXEC.BAT.
____Guided Activity: Create a CONFIG.SYS File.
____Review Questions.
____Unit Exercises.
____Documentation Research.

Place a check in front of those assigned by your instructor.

THE INVISIBLE DOS COMMAND

If you find during your work that you are entering the same commands each time you start your computer, a batch file (sometimes referred to as a BAT file) can make your computing easier and more efficient. BAT files are programs you create that can be executed from the keyboard or from within other batch files, i.e., from within an AUTOEXEC.BAT file when your computer system is booted. The commands in your BAT files respond the same way as if you were entering them from the keyboard. BAT files save you time because you only have to list the commands in the file, and then the file does the work for you.

In this unit, you will create two kinds of BAT files; one is called AUTOEXEC.BAT, and the other is referred to simply as a BAT file. An AUTOEXEC.BAT file is a special BAT file executed from your root directory when you boot up your computer. When DOS sees an AUTOEXEC.BAT file, it automatically executes the commands it finds in the file. A BAT file is created the same way as the AUTOEXEC.BAT file but does not use AUTOEXEC in its name. This way, DOS does not execute the file when you start your computer. BAT files can be executed by typing the filename and pressing <ENTER>.

BAT files can be created with a text editor or COPY CON. In this unit, you will create a BAT file with COPY CON; if you make any errors or want to add commands, you will do these tasks with EDLIN.

GUIDED ACTIVITY

CREATE THE FIRST BAT FILE: To better understand BAT files, let's make a small one. Remember, when using COPY CON you can only make corrections to the line you are on. Once you press <ENTER>, you will not be able to return to that line to make corrections. If you make an error and want to cancel COPY CON and start over, press <CTRL-C>, and the file will be cancelled.

1. Turn on the computer.
2. Enter the current date and time.
3. Place a formatted system diskette in drive A:.
4. Type **CD\DOS**
5. Press <ENTER>.
6. Type **COPY CON A:HELLO.BAT**
7. Press <ENTER>.

The ECHO OFF command will stop the commands used in your files from appearing on your screen when the file is executed, but you will see the results of the command.

8. Type **ECHO OFF**
9. Press <ENTER>.

The CLS command will clear your screen of all characters that are displayed up to the point you used the command. Sometimes, it is helpful to remove the previous instructions or clutter from the screen in order to avoid confusion.

10. Type **CLS**
11. Press <ENTER>.

The ECHO command will display the contents of a line to the screen.

12. Type **ECHO HELLO!**
13. Press <ENTER>.
14. Type **ECHO It sure is a good day to learn DOS!**
15. Press <ENTER>.
16. To end the file, press <CTRL-Z>. (F6)

17. Press <ENTER>. The HELLO.BAT file is now stored on drive A:.

Figure 7-1 shows the HELLO.BAT file.

```
C:\>CD\DOS

C:\DOS>COPY CON A:HELLO.BAT
ECHO OFF
CLS
ECHO HELLO!
ECHO It sure is a good day to learn DOS!
^Z
        1 File(s) copied

C:\DOS>
```

Fig. 7-1. Creating HELLO.BAT.

18. Type **A:HELLO**

19. Press <ENTER>.

Figure 7-2 shows the screen after executing HELLO.BAT.

20. Type **CLS**

21. Press <ENTER>.

```
HELLO!
It sure is a good day to learn DOS!
C:\DOS>
```

Fig. 7-2. Executing HELLO.BAT.

GUIDED ACTIVITY

ADD INVISIBLE LINES TO THE HELLO.BAT FILE : If you decide you need some space in your BAT file, you can insert invisible lines with the ECHO command. When you type **ECHO:** into the file, DOS will leave a blank line at that point.

1. Type **EDLIN A:HELLO.BAT**
2. Press <ENTER>.
3. Type **1L**
4. Press <ENTER>.
5. Type **4i**
6. Press <ENTER>.
7. Type **ECHO:**
8. Press <ENTER>.

9. Press <F3>.
10. Press <ENTER>.
11. Press <F3>.
12. Press <ENTER>.
13. Press <F3>.
14. Press <ENTER>.
15. Press <CTRL-Z>.
16. Press <ENTER>.
17. Type **E**
18. Press <ENTER>.

Figure 7-3 shows that four blank lines have been added to the HELLO.BAT file.

```
C:\DOS>EDLIN A:HELLO.BAT
End of input file
*1L
       1:*ECHO OFF
       2: CLS
       3: ECHO HELLO!
       4: ECHO It sure is a good day to learn DOS!
*4i
       4:*ECHO:
       5:*ECHO:
       6:*ECHO:
       7:*ECHO:
       8:*^Z
*E

C:\DOS>
```

Fig. 7-3. Revised HELLO.BAT file.

19. Type **CLS**

20. Press <ENTER>.

21. Type **A:HELLO**

22. Press <ENTER>.

Figure 7-4 shows the HELLO.BAT file, which now has four blank lines between the "HELLO!" and "It sure is a good day to learn DOS."

```
HELLO!



It sure is a good day to learn DOS!
C:\DOS>
```

Fig. 7-4. Revised HELLO.BAT file.

23. Type **CLS**

24. Press <ENTER>.

GUIDED ACTIVITY

TERMINATE A BATCH FILE: There will be times when you want to stop a BAT file. If you press <CTRL-C>, DOS will stop the BAT file and respond with the message "Terminate batch job (Y/N)?".

If you press **Y**, the BAT file will be cancelled, and the DOS prompt will return.

> **NOTE:** Because you have a small BAT file, you will have to be fast with the <CTRL-C> keystroke to stop the file.

1. Type **A:HELLO**
2. Press <ENTER>.
3. Press <CTRL-C>.

Figure 7-5 shows the screen response to the key command <CTRL-C>.

```
C:\DOS>A:HELLO
C^C
Terminate batch job (Y/N)? Y
C:\DOS>
```

Fig. 7-5. Terminating a Batch File.

4. To terminate batch job, type **Y**
5. Type **CLS**
6. Press <ENTER>.

GUIDED ACTIVITY

STOP THE BAT FILE WITH THE PAUSE COMMAND: If the file executed too fast for you and you still want to stop it, use the PAUSE command in your file. The PAUSE command will stop a BAT file to allow you the time needed to accomplish your task. For example, you might need to switch disks from one drive to another; PAUSE provides the time you need to do so.

1. Type **EDLIN A:HELLO.BAT**
2. Press <ENTER>.
3. Type **1L**
4. Press <ENTER>.
5. Type **4**
6. Press <ENTER>.
7. Type **PAUSE**
8. Press <ENTER>.
9. Type **E**
10. Press <ENTER>.

Figure 7-6 show the addition of the PAUSE command to the file.

11. Type **CLS**
12. Press <ENTER>.

```
C:\DOS>EDLIN A:HELLO.BAT
End of input file
*1L
       1:*ECHO OFF
       2: CLS
       3: ECHO HELLO!
       4: ECHO:
       5: ECHO:
       6: ECHO:
       7: ECHO:
       8: ECHO It sure is a good day to learn DOS!
*4
       4:*ECHO:
       4:*PAUSE
*E

C:\DOS>
```

Fig. 7-6. PAUSE command.

13. Type **A:HELLO**
14. Press <ENTER>.
15. Press <ENTER> when you are ready to continue.
16. Type **A:HELLO**
17. Press <ENTER>.
18. Press <CTRL-C>.

Figure 7-7 shows the screen request for termination.

19. To terminate the batch job, type **Y**
20. Type **CLS**
21. Press <ENTER>.

```
HELLO!
Strike a key when ready . . . ^C

Terminate batch job (Y/N)? Y

C:\DOS>
```

Fig. 7-7. Termination message.

GUIDED ACTIVITY

CREATE AN AUTOEXEC.BAT FILE: The AUTOEXEC.BAT file is a batch file placed in the root directory of your disk. Remember, when DOS starts your computer at boot-up, it looks at the root directory, and if it finds a file named AUTOEXEC.BAT, it will execute the file and follow the commands within the file. You are going to add the DATE and TIME commands, remove the PAUSE command, and place the AUTOEXEC.BAT file in the root directory of your disk.

1. Type **COPY CON A:AUTOEXEC.BAT**
2. Press <ENTER>.
3. Type **ECHO OFF**
4. Press <ENTER>.
5. Type **CLS**
6. Press <ENTER>.

7. Type **DATE**
8. Press <ENTER>.
9. Type **TIME**
10. Press <ENTER>.
11. Type **CLS**
12. Press <ENTER>.
13. Type **ECHO HELLO!**
14. Press <ENTER>.
15. Type **ECHO:**
16. Press <ENTER>.
17. Press <F3>.
18. Press <ENTER>.
19. Press <F3>.
20. Press <ENTER>.
21. Press <F3>.
22. Press <ENTER>.
23. Type **ECHO It sure is a good day to learn DOS!**
24. Press <ENTER>.
25. Press <CTRL-Z>.
26. Press <ENTER>. The AUTOEXEC.BAT file is now stored on the disk in drive A:.

Figure 7-8 shows the AUTOEXEC.BAT file.

```
C:\DOS>COPY CON A:AUTOEXEC.BAT
ECHO OFF
CLS
DATE
TIME
CLS
ECHO HELLO!
ECHO:
ECHO:
ECHO:
ECHO:
ECHO It sure is a good day to learn DOS!
^Z
        1 File(s) copied

C:\DOS>
```

Fig. 7-8. AUTOEXEC.BAT file.

GUIDED ACTIVITY

REBOOT THE COMPUTER WITH AUTOEXEC.BAT:

1. With the system disk in drive A: reboot the computer.
2. Enter the current date.
3. Press <ENTER>.
4. Enter the current time.
5. Press <ENTER>.

Your screen should greet you with the results of your AUTOEXEC.BAT file; see Fig. 7-9.

6. Type **CLS**
7. Press <ENTER>.

```
HELLO!



It sure is a good day to learn DOS!
A>
```

Fig.7-9. AUTOECEC.BAT file.

CONFIGURATION SYSTEM FILE

The CONFIGURATION SYSTEM file, or CONFIG.SYS as it is most commonly called, is a file DOS uses at start-up to configure your computer system. The CONFIG.SYS file can change your DOS system in certain ways. When you start your system DOS, looks in the root directory to see if you have a CONFIG.SYS file; if it finds that file, it will carry out the commands in the file. The commands you can have in a CONFIG.SYS file are BREAK, BUFFERS, COUNTRY, DEVICE, FCBS, FILES, LASTDRIVE, and SHELL.

In this unit you are going to deal with the BUFFERS and FILES commands since these commands are the ones most often changed when setting up application programs. If you are interested in the remaining commands, refer to your *Disk Operating System Reference Manual* for their explanation and use.

BUFFERS COMMAND

The BUFFERS command allows you to tell DOS how many buffers you want. A buffer is block of 528 bytes of memory that DOS can use to store data so it won't

have to write data to and read data from your disks. You can have from 1 to 99 buffers with DOS 3.X. The advantage of buffers is that they speed up your computer operation; a disadvantage is that they use up the memory you might need for programs. The amount of memory available in your computer and the requirements of your application programs will determine the number of buffers you will use. The minimum number of buffers is 1, and the maximum number is 99, with a default value set at 2 for a PC and at 3 for a PC AT.

FILES COMMAND

The FILES command provides you with the ability to tell DOS how many files it can use at one time. Some application programs will require more than others. The minimum number of eight files is set by the DOS default setting. DOS allows the maximum number of files to be 255.

GUIDED ACTIVITY

CREATE A CONFIG.SYS FILE WITH COPY CON:

1. Type **COPY CON A:CONFIG.SYS**

2. Press <ENTER>.

> **NOTE:** Some application programs require different settings for the buffers and files .

3. Type **BUFFERS=16**

4. Press <ENTER>.

5. Type **FILES=20**

6. Press <ENTER>.

7. Press <CTRL-Z>. (F6)

8. Press <ENTER>.

Figure 7-10 shows the CONFIG.SYS file.

9. Type **DIR A:**

10. Press <ENTER> to view the new CONFIG.SYS file on your disk in drive A:.

If you want to take advantage of the CONFIG.SYS file, you must reboot your computer. DOS looks for the CONFIG.SYS file only at start-up.

```
A:\>COPY CON A:CONFIG.SYS
BUFFERS=16
FILES=20
^Z
        1 File(s) copied

A:\>
```

Fig.7-10. CONFIG.SYS file.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several BATCH File DOS commands. We have tried to cover the most commonly used Batch Files. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`COPY CON C:\AUTOEXEC.BAT`	Create an AUTOEXEC.BAT file for the root directory of drive C:.
`COPY CON C:\CONFIG.SYS`	Create a CONFIG.SYS file for the root directory of drive C:.

REVIEW QUESTIONS

1. In what directory do you place an AUTOEXEC.BAT file?
2. How would you stop a batch file during execution?
3. What procedure would you follow to insert a blank line within a batch file?
4. How many bytes of memory do five buffers use?

UNIT EXERCISES

1. Create an AUTOEXEC.BAT file that will display the current subdirectory and list the files within the subdirectory.

DOCUMENTATION RESEARCH

1. Research the *IBM Disk Operating System Version 3.3 Reference* manual for DOS commands and parameters not used in class.

APPLICATION E

PART 1

CREATING BATCH FILES:

A. Create a BATCH file that will make the subdirectories shown in the diagram on page 131.

PART 2

USING BATCH FILES TO COPY FILES TO SUBDIRECTORIES:

A. Create a BATCH file which copies all the months of the year to subdirectories \ACCT\GL and \ACCT\AR\AP.

PART 3

BATCH FILES FOR REMOVING SUBDIRECTORIES:

A. Create a BATCH file that will remove all files and subdirectories that were made in Part 1 and Part 2.

UNIT 8

DOS SAFETY COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. define the key component in any disaster recovery plan;
2. explain the use of the BACKUP and RESTORE commands in contrast to the COPY and DISKCOPY commands;
3. list the different parameters appropriate to the BACKUP and RESTORE commands;
4. list the two primary copy-protection methods used by software manufacturers and comment on how they affect the use of the BACKUP and RESTORE commands;
5. list the three DOS commands that interfere with the BACKUP and RESTORE commands;
6. discuss the need for a periodic system backup;
7. explain the two uses of the ATTRIB command;
8. list at least two uses of a write-protected file;
9. explain and use the RECOVER command;
10. explain and use the XCOPY and REPLACE commands.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. ATTRIB
2. BACKUP
3. RECOVER
4. REPLACE
5. RESTORE
6. XCOPY.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following terms listed in the Computer Terms Dictionary (Appendix A): alternative processing strategies, attribute bits, backup disks, boilerplate documents, copy-protection, disaster recovery plan, enterprise, hard-disk crash, manual backup systems, read-and-write, read-only, redundancy, and write-protect.
____Guided Activity: Use the Backup and Restore Process.
____Guided Activity: Use the Partial Restore Process.
____Guided Activity: Mark Files Read-Only with the ATTRIB Command.
____Guided Activity: Use the XCOPY Command.
____Guided Activity: Use the REPLACE Command.
____Review Questions.
____Unit Exercises.
____Documentation Research.

Place a check in front of those assigned by your instructor.

HOW TO SMILE WHEN DISASTER STRIKES

Disaster recovery planning should be a basic part of the planning process for any data processing center. Microcomputer users tend to ignore this aspect of planning and implementation until disaster strikes. The most common function in a **disaster recovery plan** is scheduling regular system and data backups.

A complete disaster recovery plan must include such topics as **alternative processing strategies**, testing, security, **manual backup systems**, cost effectiveness, and impact on the **enterprise** (business or department). Our scope of concern in this unit centers on backup systems as provided by PC DOS and MS-DOS, plus an examination of the RECOVER command.

Two additional factors enter into any backup strategy. First is the consideration of the kind of medium to be used. Microcomputer systems use several types of media,

including tape, disk, and cartridge backups. The second consideration is whether to do a complete backup or to back up only modified files.

PC DOS versions 3.3 and lower do not support disk-to-tape backups, so we limit the discussion to backups onto floppy diskettes. With DOS 2.0, the backup process permits the backup of files from a hard disk to floppy diskettes only. DOS 3.1 added the capability of backing up from

1. one diskette to another diskette;
2. a diskette to a hard disk;
3. one hard disk to another hard disk.

Users who purchase microcomputer tape backup systems receive software to facilitate backup to tape or cartridge, even though DOS does not provide that capability.

WHY DISASTER MAY STRIKE

In Unit 1 we discussed the care and handling of diskettes. From that discussion, it should be clear that floppy disks may go "bad" or be damaged. Even expert care can sometimes be voided by accidentally stacking books on a disk, spilling liquid on a disk, or dropping the disk on a dusty surface. A hard disk appears to be more protected. It is encased in a metal enclosure inside the PC's system unit. Unfortunately, anything mechanical will eventually wear down. A hard disk is no exception.

A **hard-disk crash** is every user's greatest fear--yet every hard disk will crash. It is only a matter of time. The term *hard-disk crash* means that the read-write head in the hard-disk unit comes into destructive contact with the disk platters. A hard disk is composed of several rigid platters, not unlike a series of rigid diskettes. When the head gouges or scratches the recording surface of a platter, data is lost, and that area can no longer store data. If the damage occurs on the area reserved for track zero (where the directory and file allocation tables reside), the hard disk becomes unusable.

What causes a hard-disk crash or head crash? The two main causes are the following:

1. the read-write head mechanically fails and bumps into the recording surface;
2. the system unit is jarred or moved while the read-write head is in motion, thus causing the head to damage the recording surface.

In normal operation, the read-write head floats over the recording surface. If the head physically comes into contact with the platter, damage occurs. The IBM Diagnostic diskette, which is provided with each new system unit, includes a program to park the head of a hard-disk unit. Before a hard-disk system unit is moved, the read-write head should be parked. The software moves the head to a

position above the platters and beyond the last track or cylinder. In that position vibrations from moving the unit are less likely to cause damage.

All hard-disk failures are not the result of a read-write head crash. Additional problems include

1. damage from power surges;
2. damage from radio frequency interference (RFI);
3. damage from static electricity.

THE DOS BACKUP COMMAND

The COPY and DISKCOPY commands may be used if a backup is limited in scope. For instance, if all data is on a disk or two, either COPY or DISKCOPY will work. Because the DISKCOPY command is intended for floppy disk copying only, it cannot be used to back up files on a hard disk.

The COPY command, on the other hand, can be used to back up data from a hard disk. Two limitations exist for this type of backup:

1. Any diskette is limited in the number of files it can store in its directory. To be technical, the limitation is on the number of files that may be stored in the root directory. The number may be increased by using one or more subdirectory entries in the root directory. Each subdirectory is unlimited in the number of files it can store. Each file on the disk, the disk volume label, each subdirectory, and each hidden file count toward the file directory table limit.

2. Any diskette is limited in the amount of bytes it can store. A diskette's storage capacity is further reduced by the space occupied by the boot record, the file allocation table (FAT), and the directory table. For example, a 360K floppy can really only store 354K in data.

Figure 8-1 shows the various disk types, sectors per track, number of directory entries, and other diskette data. When either of these limits is exceeded, the copying process terminates with the error message insufficient disk space, and results in incomplete file copying.

Disk Type	Storage	Size	Sectors	FAT Sectors	DIR Sectors	Directory Listings
SS 2D (DD)	160KB	5¼	8	2	4	64
SS 2D (DD)	180KB	5¼	9	4	4	64
DS 2D (DD)	320KB	5¼	8	2	7	112
DS 2D (DD)	360KB	5¼	9	4	7	112
DS 2D (DD)	720KB	3½	9	6	7	112
DS 4D (HD)	1.2MB	5¼	15	14	14	224
DS 4D (HD)	1.44MB	3½	18	18	14	224
Hard disk	10MB		17	16	32	512
Hard disk	20MB		17	80	32	512
Hard disk	30MB		17	120	32	512

Note: SS = single-sided, DS = double-sided, 2D = double-density, 4D = quad-density.

Fig. 8-1. Data on various diskette types.

The COPY command might still be useful if a strategy is developed that would group files with similar extensions or similar file specifications using global wildcard characters (such as `COPY *.DOC`, `COPY *.WKS`, `COPY *.DBF`, `COPY LETTER??.*`). This process obviously eliminates typing a separate COPY command line for each and every file on the disk. The schema reduces the number of commands issued to back up the entire system and, if grouped properly, would allow multiple diskettes to be used in the process. If the files are not grouped appropriately, you may run out of space on a given diskette halfway through copying a specified group of files. If that occurs, you must rethink the file groupings or go back and copy files one by one to the target disk.

When backing up a hard disk or subdirectory that requires more than two to three diskettes, the COPY command becomes an awkward substitute for the DOS BACKUP command.

NOTE: One distinct advantage of the BACKUP command is its ability to divide a large file over two or more diskettes. The COPY command cannot copy a 420K file to a 360K target disk. The BACKUP command will back up that file to two diskettes. Of course, it cannot be accessed or used until it is restored to a single disk using the RESTORE command.

BACKUP is an external DOS command created specifically to handle hard-disk system backups. Because it is external, either the BACKUP.COM file from the DOS diskette must be available on the default drive, or the drive and path to the BACKUP.COM command must be specified.

The syntax of the command is

```
[d:][path]BACKUP s:[filename.ext] t:[filename.ext]
```

where s: represents the source of the files to be backed up and t: represents the drive on which the duplicate files will be created. In its simplest form, all files on drive A: may be backed up to a disk in drive B: by invoking this command:

```
BACKUP A:*.* B:
```

> **NOTE:** Although MS-DOS 3.3 and PC DOS 3.3 follow the above syntax, some versions of DOS provided by clone manufacturers would require that this command be entered as `BACKUP A:*.* B:*.*`.

In this form, the BACKUP command creates a backup of files in only the current or specified directory. It does not automatically back up files in any other directory, nor in the subdirectories below the current or specified directory.

One very helpful improvement with the 3.x version of BACKUP is its ability to format a disk, using the /F parameter, during the backup process. Unlike earlier versions of DOS, you no longer need to format a group of diskettes before executing the FORMAT command. If you use the /F parameter, the BACKUP command formats the disk and then backs up files to the disk.

Seven parameters (including the /F parameter) may be used with the BACKUP command. These parameters may be combined on a single command line if needed.

/A	Appends backed-up files to files already on the **backup disk**.
/D:mm-dd-yy	Backs up files that have been modified on or after a stipulated date.
/F	Formats a disk and then backs up files from the source to the newly formatted target disk.
/L	Creates a backup log in the root directory. The log is stored in a file named BACKUP.LOG and contains the date, time, filename, and diskette number of the backup.
/M	Backs up all files modified since the last backup; that is, those with an archive bit turned on.
/T:hh:mm:ss	Backs up files that have been modified on or after a stipulated time.
/S	Backs up all files in the current or designated directory plus all subdirectory files below the current or specified directory.

The last parameter facilitates the backup of an entire hard disk. If the current directory is the root directory and the /S parameter is used, every file in every directory or subdirectory on the fixed disk will be backed up.

To back up all files to a floppy diskette in the A: drive, the form of the command would be

```
BACKUP C:*.* A:/S
```

If we sought to back up only files modified since the last backup, we could add one more parameter.

```
BACKUP C:*.* A:/S/M
```

If this new form of the command is executed in the root directory, it will cause all directories and subdirectories to be searched for files modified since the last backup. Those files will then be placed on the backup disk. Whenever a file is backed up, one of the **attribute bits** in the directory table is altered to indicate that the process has taken place. If that file is subsequently modified, the flag is changed to indicate that no backup has taken place since the last modification. When using the /M parameter, DOS looks only for flagged files, i.e., those not marked as backed up in their current form. Those files only are then transferred to the backup disk. During the transfer process, the file is then altered to indicate it has been backed up.

The date parameter (/D) can be useful only if you seek to back up files created or modified either on or since a specified date. If a periodic backup program has been implemented, the date parameter could be used by entering the day after the last backup.

In the situation where all directories and subdirectories are to be searched, and all files created or modified since 15 August 1986 are to be backed up, the command form would be

```
BACKUP C:*.* A:/S/D:8-15-86
```

In each of the last three examples, we have assumed that the default drive is C: and that the system is logged on the root directory. If the default directory were not the root directory, only files in the current directory and its subdirectories would be backed up. The \ character represents the root directory; when it is combined with the /S parameter, it forces the backup process to begin in the root directory and proceed to back up each subdirectory on the disk. With the date option, the modified command would be

```
BACKUP C:\*.* A:/S/D:8-15-86
```

The /T parameter enables you to back up all files with creation times matching or after the stipulated time. For instance, the following command would back up all files with a creation time matching or after 1:15 p.m.:

```
BACKUP C: A:/S/T:01:15p
```

You must designate an afternoon time (p.m.) by using the "p". If the time stipulated is in the morning, you do not need to add the "a". So to back up all files created since 10:01 a.m., the command would be

```
BACKUP C: A:/S/T:10:01
```

The date and time parameters (respectively, /D and /T) may be used in combination. If you wish to back up only those files created at 11:25 a.m. on January 1, 1989, or after, you could use the following command form:

```
BACKUP C: A:/S/D:01-01-89/T:11:25
```

The /L parameter creates a text file that indicates the full filename and path of backed-up files and which backup disk that file appears on. BACKUP.LOG also includes the date and time (using a 24-hour clock) of the backup. You can use the TYPE command to view the contents of this file.

BACKUP COMMAND ERROR CODES

Any disaster recovery plan has as its foundation the backup of all files stored or used by the user. If those backups are faulty, the entire recovery plan is disastrously flawed. DOS provides the following five exit error codes to assist the operator in determining whether the BACKUP command has terminated successfully or unsuccessfully. These codes can be accessed using the IF ERRORLEVEL batch processing subcommand. Otherwise, they are not displayed on your monitor screen.

0 BACKUP command completed normally.
1 DOS found no files to back up.
2 Due to file sharing conflicts on a local area network (LAN), some files were not backed up.
3 BACKUP command was terminated by the user (using the <CTRL-BREAK> Key combination).
4 Terminated because of other error.

WHAT TO DO AFTER THE CRASH

The BACKUP command has a twin called RESTORE. After a hard-disk failure or any other disaster, the RESTORE command is used to transfer files from the backup disks to a new hard disk or reformatted disk. RESTORE is an external DOS command. The syntax of the command is

```
[d:][path]RESTORE s: [t:][filename.ext]
```

where s: is the drive containing the files to be restored and t: is the target drive.

In its simplest form, a restoration of all files on backup disks in drive A: to drive B: would be accomplished with this command:

```
RESTORE A: B:*.*
```

The RESTORE command uses eight parameters. All but two of these are new with DOS 3.x.

/A:mm-dd-yy	Restores all files on or after the specified date.
/B:mm-dd-yy	Restores all files on or before the specified date.
/E:hh:mm:ss	Restores all files modified at or earlier than the specified time.
/L:hh:mm:ss	Restores all files modified at or later than the specified time.
/M	Restores only files modified or deleted since the last backup.
/N	Restores only files no longer on the target disk.
/P	Causes the RESTORE command to prompt you before it restores files that are marked read-only or that have been modified since the last backup. The prompt allows the user to choose to restore or not, whichever is most appropriate.
/S	Restores all files in the current or designated directory, plus all subdirectory files below the current or specified directory.

The files on the source diskette must have been placed there by the BACKUP command. These commands are Siamese twins; they work only in conjunction with each other. Here is another example using the /S parameter. Let's assume several points:

1. The source backup disk is in drive A:.
2. You seek to restore files to the hard disk, drive C:.
3. The default drive is the root directory on drive C:.
4. Files are to be restored to all subdirectories under the root directory.

With these assumptions, the correct form of the command is

```
RESTORE A: C:*.*/S
```

If the default drive and current directory were anything other than those stipulated in the assumptions, the command would need to be revised to

```
RESTORE A: C:\*.*/S
```

You could restore only files that had been deleted after the backup by using the command form

```
RESTORE A: C:\*.*/N
```

With eight different parameters, you have a number of options for restoring files. The parameters may be combined, if needed. For instance, if you wish to restore (to all subdirectories under the root) only files created since 10 a.m. on July 1, 1988, you would use the command form

```
RESTORE A: C:\*.*/S/D:07-01-88/T:10:00
```

The RESTORE command uses four exit codes similar to those outlined in the section on the BACKUP command. These codes can be accessed using the IF ERRORLEVEL batch processing subcommand. Otherwise, they are not displayed on your monitor screen.

0 RESTORE command completed normally.
1 DOS found no files to restore.
3 RESTORE command was terminated by the user (using the <CTRL-BREAK> Key combination).
4 Terminated because of other error.

CAVEATS TO THE DOS BACKUP AND RESTORE COMMANDS

The BACKUP command is not the same as the COPY command. Files created by the BACKUP command cannot be accessed or used until they have been restored. In fact, the BACKUP command only creates two files on the backup diskette. The two files are

BACKUP.999 and
CONTROL.999

The 999 is replaced by the number representing the sequential order of the backup disk. If the backup requires one disk, the files will be named

BACKUP.001 and
CONTROL.001

If the backup requires three diskettes, each backup disk will hold two files with the following filenames:

Diskette 1	Diskette 2	Diskette 3
BACKUP.001	BACKUP.002	BACKUP.003
CONTROL.001	CONTROL.002	CONTROL.003

Keep in mind that the COPY and BACKUP commands serve different purposes; they are not interchangeable.

The RESTORE command does not incorporate the FORMAT command, as the BACKUP command does beginning with DOS 3.x. The target disk used with the RESTORE command must already be formatted.

#2 Also, be sure to number the diskettes created via the BACKUP command. They must be numbered because the RESTORE command requires that the restoration process use the diskettes in the same order in which they were created. Keep in mind that it will take twenty-five to thirty 360K diskettes to back up each 10MB of hard-disk storage.

> **NOTE:** To be exact, each 360K diskette really holds only 354K (362,496 bytes) of data storage space (after deducting the sectors for FAT and the directory table). If you had exactly 10MB of data on your hard disk, it would take 28.9265536723 diskettes to store that data.

The BACKUP command will erase all files off a diskette unless the /A parameter is used. This is cause for caution. The backup diskettes should be clearly labeled as such. Data on any diskette mistakenly used as a backup diskette will be lost. Further, it is wise to have a minimum of two completely separate sets of backup diskettes. Most seasoned computer users have been through the horrendous experience of having a read error on a hard disk, having a single set of backup diskettes, and finding the set flawed. The greater the number of backups, the greater the safety in case of disaster.

To maximize the chances of having a recent, usable set of backup diskettes available when necessary, you must have greater **redundancy**. Redundancy requires that hardware, software, and data be duplicated. True redundancy in a microcomputer business setting would require

1. a backup microcomputer available for each user if his or her original machine malfunctioned;
2. an archival set of program diskettes available to reinstall;
3. a recent complete backup of all data created by the user.

A TYPICAL BACKUP SCHEDULE

Figure 8-2 shows a typical backup schedule for a single-user PC with a medium level of activity. In a high-volume business setting or a multiuser setting (where PCs are networked), this schedule would be too infrequent.

The backup schedule displayed in Fig. 8-2 uses multiple sets of backup diskettes plus partial and complete backups. This particular schedule uses four sets of backup diskettes labeled A, B, C, and D. The C and D diskettes are used only for

complete backups. Sets A and B are used to back up modified files (in between each Friday full-system backup).

In a typical office setting, the backup would only cover specific hard-disk subdirectories containing data. If you have an archival set of program disks for your application programs, those files do not need to be backed up. The original program disks or an archival copy provides the redundancy needed.

The schedule assumes that the /S parameter is used when needed to back up all subdirectories containing data. The strategy follows these guidelines:

1. Make a full backup every Friday with alternating sets of diskettes.
2. Make a backup of modified files on Monday through Thursday.
3. The Monday and Tuesday backup of modified files does not use the /A parameter. This overwrites all files on the disk sets except files altered since the full backup on the previous Friday.
4. The Wednesday and Thursday backup of modified files uses the /A parameter. This adds additional altered files to disk sets A and B, respectively, yet preserves the modified files backed up on Monday and Tuesday.

September 1990

Single User Backup Schedule

SUN	MON	TUE	WED	THU	FRI	SAT
						1
2	3 BACKUP/M to set B	4 BACKUP/M to set A	5 BACKUP/A/M to set B	6 BACKUP/A/M to set A	7 Full BACKUP to set C	8
9	10 BACKUP/M to set B	11 BACKUP/M to set A	12 BACKUP/A/M to set B	13 BACKUP/A/M to set A	14 Full BACKUP to set D	15
16	17 BACKUP/M to set B	18 BACKUP/M to set A	19 BACKUP/A/M to set B	20 BACKUP/A/M to set A	21 Full BACKUP to set C	22
23	24 BACKUP/M to set B	25 BACKUP/M to set A	26 BACKUP/A/M to set B	27 BACKUP/A/M to set A	28 Full BACKUP to set D	29
30						

August

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

October

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Fig. 8-2. A typical full and modified-only backup schedule.

In addition to the sets of backup diskettes used in the schedule above, most businesses would expect to perform separate weekly, monthly, and year-end backups. These are completely separate from the backup sets A, B, C, and D used above. With financial applications it is even more critical to create backups for regular accounting periods. You should create an endoftheweek backup (one per week in the month), a monthly backup (one per month in the year), and a year-end backup (could be a fiscal or calendar year backup--though the former is more usual).

Three DOS commands exist that can play havoc with the backup and restore process. The first two are JOIN and SUBST, DOS commands born with the advent of DOS version 3.1. They should never be invoked while the BACKUP or RESTORE command is to be used. The third is ASSIGN. Both ASSIGN and SUBST may confuse either DOS or the user as to the true drive designation to be used in a backup and restore process. The JOIN command can confuse the tree structure, thus making it invalid after a BACKUP and RESTORE procedure.

There is one additional class of problem software that loves to decimate carefully planned and executed backups. It is the group of handy little memory-resident programs that includes clock display software, background communications programs, print spoolers, note pads, telephone dialers, screen savers, and much more. These programs increase productivity, but may cause untold havoc when running concurrent with DOS's BACKUP and RESTORE commands. Memory-resident programs also cause problems with proprietary tape backup software such as utilized by the Tallgrass Technologies tape backup system. If you are getting ready to back up or restore your files, first reboot the system without loading any memory-resident programs!

COPY-PROTECTION AND THE BACKUP PROCESS

One final caveat for the user. Although several major manufacturers have removed **copy-protection** from their software, the nuisance remains on some software. To understand the impact on backup procedures, it is important to understand how copy-protection works. On a non-copy-protected disk, the DOS DISKCOPY command creates a duplicate disk by reading each track in sequence and copying the data to the corresponding track on the source disk. The COPY command copies a file sector by sector (or allocation unit by allocation unit) from one disk to another. With the COPY command, there is no guarantee that the file will be stored on the target disk at the same physical location--the same track or sector as the file occupied on the source diskette.

Copy-protection thwarts copying by any one of a number of methods, including rearranging the order in which the tracks are recorded, recording between tracks, recording tracks backward, using spiral rather than concentric tracks, and deliberately writing or creating bad sectors that must exist in specific locations before the disk is readable.

Typically, two methods of copy-protection persist:

1. use of a key diskette;
2. use of an install procedure that permits a limited number of installs of copy-protected files.

Manufacturers using the first copy-protection method keep all the "protected" files on the key diskette. The key diskette must be in one of the floppy disk drives for the copy-protected software to work. With this scheme, the files to be placed on a hard disk are typically not copy-protected. This method does not affect the use of the BACKUP and RESTORE commands, unless you attempt to back up and restore the key diskette.

The latter method installs "protected" files on the hard disk. These files cannot be copied using the DOS COPY, DISKCOPY, or BACKUP command. Some copy-protection schemes create hidden and system files in a hidden directory. In essence, if you have backed up your files and the hard disk crashes, there is no way to restore the copy-protected files from the backup disks. Because a limited number of installs are allowed--usually one or two--a hard-disk crash means you have lost one install. After a hard-disk failure, only two options exist: try to uninstall the copy-protected programs (often not a real possibility) or contact the software manufacturer and appeal to its mercy.

The point of this discussion? Do not expect the BACKUP and RESTORE commands to provide a usable disaster recovery system for copy-protected software. In fact, you can anticipate an 80% chance that trying to back up or restore copy-protected software will blow your software away. One popular software program recently eliminated copy-protection--but only after numerous complaints from users. Their copy-protection scheme caused the program diskette to erase automatically if the hidden files were flagged by using the DOS BACKUP command.

BACKUP--A WEARISOME EXERCISE

One of the dreariest tasks for most computer users is backing up files. It is tedious to use the DOS BACKUP and RESTORE commands. Other vendors have entered the arena with software products that provide a speedier backup than the standard DOS commands. Examples are programs such as FastBack and Back-It. Part of their speed is due to programming that causes the disk drives to continue spinning while new diskettes are inserted (no time is lost getting the drive up to speed). Further, they will alternate writing to drive A: and B:. This gives the user disk swapping time while the backup utility writes to the other drive.

Tape backup systems are even more efficient because they reduce user interaction in the backup process. Rather than swapping 25 to 30 diskettes in and out of the drive to back up 10MB of data, a tape backup system stores that much data on one tape. Some systems can store up to 60MB or more of data on one tape.

There are options for the microcomputer user beyond what is offered by MS-DOS and PC DOS. However, whichever system is used, the crucial guideline is that periodic backups are essential. Backups should occur as often as dictated by the maxim "Back up everything you cannot afford to lose!" If you cannot afford to lose a day's work, then your backup should be done at least daily. Don't let a tedious task deter you from having the security of a complete system backup.

GUIDED ACTIVITY

USE THE BACKUP AND RESTORE PROCESS: In this Guided Activity, you will use the BACKUP and RESTORE commands to create a backup of files from the West Student Data Disk, and subsequently to restore the files.

One caution--the BACKUP and RESTORE commands do not like memory-resident programs. That includes keyboard enhancers, calculators, notepads, etc. Even memory-resident programs such as screen savers that blank the screen after a predefined time period) will play havoc with BACKUP and RESTORE. Be sure that no memory-resident programs are present before you begin this exercise.

1. Log on to the root directory of your hard disk.

2. Type **MD WEST** and press <ENTER> to create a new directory on your hard disk.

3. Type **CD WEST** and press <ENTER> to change the default directory to the newly created \WEST subdirectory.

4. Place the West Student Data Disk in drive A:.

5. Type **COPY A:*.*/V** and press <ENTER> to copy all files from the West Student Data Disk to the newly created \WEST subdirectory.

6. Remove the disk from drive A:.

7. Type **CLS** and press <ENTER>.

8. Label a formatted disk as Backup Disk 01.

9. Type **BACKUP C:*.* A:** and press <ENTER>.

10. When prompted to "Insert backup diskette 01 in drive A:", place the formatted disk labeled Backup Disk 01 in drive A:.

Notice the screen warning that "Files in the target drive A:\ root directory will be erased".

11. Press any key.

The screen should echo the following message:

*** Backing up files to drive A: ***
Diskette Number: 01

When all nine files contained on the West Student Data Disk are backed up, your screen will display the messages shown in Fig. 8-3.

```
C:\WEST>BACKUP C:*.* A:

Insert backup diskette 01 in drive A:

Warning! Files in the target drive
A:\ root directory will be erased
Strike any key when ready

*** Backing up files to drive A: ***
Diskette Number: 01

\WEST\TOUCHY.TXT
\WEST\TECHNO.TXT
\WEST\TRUCKIN.TXT
\WEST\DEGAS.BDC
\WEST\DEGAS.DOC
\WEST\EAGLE.BDC
\WEST\EAGLE.DOC
\WEST\OVERDUE.DOC
\WEST\PCAT.DOC

C:\WEST>
```

Fig. 8-3. Screen activity resulting from executing the BACKUP command.

12. When the DOS prompt appears again (C:\WEST>), type **DIR A:** and press <ENTER>.

Note that only two files are listed on your backup disk. BACKUP.001 stores the data for all the files while CONTROL.001 stores the volume number (i.e., 01), filenames, and other related data used by the RESTORE command. Remember that the BACKUP command creates a mirror image that is readable only by the RESTORE command.

13. Type **CLS** and press <ENTER>.

14. Type **TYPE A:BACKUP.001** and press <ENTER>.

The screen will display part of the text from the backed-up files.

15. Type **TYPE TOUCHY.TXT** and press <ENTER>.

The results are the same and are displayed in Fig. 8-4.

```
C:\WEST>TYPE A:BACKUP.001
Electroactive polymeric material that is capable of
sensing shape, pressure, and texture.  That's the
definition, but what does it define?  The answer is
robot fingers!

Georgia Tech's Energy and Materials Sciences Lab
has been conducting research on the creation of
"touchie-feelie" robots since 1986.  They
anticipate that robots with a "soft touch" might
someday replace humans in NASA's space program.
Heck, they take the fun out of everything.
C:\WEST>TYPE TOUCHY.TXT
Electroactive polymeric material that is capable of
sensing shape, pressure, and texture.  That's the
definition, but what does it define?  The answer is
robot fingers!

Georgia Tech's Energy and Materials Sciences Lab
has been conducting research on the creation of
"touchie-feelie" robots since 1986.  They
anticipate that robots with a "soft touch" might
someday replace humans in NASA's space program.
Heck, they take the fun out of everything.
C:\WEST>
```

Fig. 8-4. Using the TYPE command on BACKUP.001 and TOUCHY.TXT.

The TYPE command will read a file and display it to the screen until an end-of-file (EOF) character is found. The BACKUP command copied all the files in the C:\WEST subdirectory into a single file--including each file's EOF character. The first file on the disk is TOUCHY.TXT--that is why you can only view the text from that single file when executing the **TYPE A:BACKUP.001** command.

16. Remove the Backup Disk 01 from drive A:.

17. Be sure the default drive and default directory are C:\WEST.

18. Type **DIR *.*** and press <ENTER>.

19. The screen should list the 11 files, 9 of which were copied from the West Student Data Disk to the C:\WEST subdirectory earlier in this exercise. If these files are not listed, use the CD command to change to the C:\WEST subdirectory before proceeding.

> **NOTE:** Of the 11 files listed by the DIR command, 2 are really directory listings. They are represented by the ..<DIR> and .<DIR> listings. The former notation represents the parent directory (the directory immediately above the current one), and the latter represents the current directory.

20. Type **DEL *.*** and press <ENTER>.

21. When prompted "Are you sure (Y/N)?", type **Y** and press <ENTER>.

22. Type **DIR** and press <ENTER>.

The new subdirectory C:\WEST has no files in it, but it does include the two directory listings (.<DIR> and ..<DIR>).

23. Type **CLS** and press <ENTER>.

In the following steps, you will restore the backed-up files to the C:\WEST subdirectory.

24. Type **RESTORE A: C:*.*** and press <ENTER>.

25. When prompted to "Insert backup diskette 01 in drive A:", place the Backup Disk 01 in the A: drive, and then press any key.

The screen displays a message indicating the date the backup was created. As the files are restored, they are listed on the screen. The list of backed-up files looks like that shown in Fig. 8-5.

```
C:\WEST>RESTORE A: C:*.*

Insert backup diskette 01 in drive A:
Strike any key when ready


*** Files were backed up 05-03-1989 ***

*** Restoring files from drive A: ***
Diskette: 01
\WEST\TOUCHY.TXT
\WEST\TECHNO.TXT
\WEST\TRUCKIN.TXT
\WEST\DEGAS.BDC
\WEST\DEGAS.DOC
\WEST\EAGLE.BDC
\WEST\EAGLE.DOC
\WEST\OVERDUE.DOC
\WEST\PCAT.DOC

C:\WEST>
```

Fig. 8-5. Screen response when using RESTORE command.

26. Type **DIR** and press <ENTER>. All files from the West Student Data Disk should again appear on the disk.

ADDITIONAL NOTES ON RESTORE

A few supplemental notes on the RESTORE command may be helpful. First, a single file or group of files may be restored using the RESTORE command. Even though the previous Guided Activity focused on the total restoration of files, you can select the option of partially restoring from one or more backup diskettes.

In the previous exercise, the Backup Disk 01 includes the TECHNO.TXT file. That single file could be restored by using the command line

```
RESTORE A: C:TECHNO.TXT
```

Groups of files may be restored using the DOS wildcard characters. If a backup diskette (in drive A:) included an assortment of worksheet files (.WS extension), backup worksheet files (.BWS extension), text files (.TXT extension), and document files (.DOC extension), a partial restoration of all document files (to drive B:) could be accomplished using the command line

```
RESTORE A: C:*.DOC
```

The RESTORE command has a minor idiosyncrasy related to subdirectories. If you have organized your hard disk (or a floppy disk) into subdirectories, you can only restore files to the subdirectory from which they were originally backed up. This is really a minor disadvantage in most cases. However, if you were switching from a 10MB hard disk to a 20MB drive and wished to reorganize the subdirectories on the new drive, the process would be awkward. You would need to

1. back up the 10MB hard disk;
2. remove the 10MB drive, and replace it with the 20MB drive;
3. restore all files to the 20MB drive into the same subdirectories they occupied on the old drive;
4. create the new subdirectory structure;
5. copy files into the new subdirectories;
6. delete files from the old subdirectories;
7. remove old subdirectories that are empty of files.

One nice feature--the RESTORE command will recreate subdirectories if they were lost due to accidental erasure or when switching to a new hard disk. If the subdirectory existed when the backup was made, and does not exist on the target drive, the subdirectory will be recreated during the restoration process.

HISTORICAL NOTE: Under certain circumstances, the BACKUP command provided in DOS version 2.x would substitute a hexadecimal 0 for one of the filename characters on the target diskette. This error produced a set of flawed backup diskettes that would not work with the RESTORE command. Thus, the time and effort used to back up data with version 2.1 was wasted--because the data could not be restored. This bug was cured with DOS version 3.x. Another problem is that if you back up files with one version of DOS and then try to restore with another, it probably will not work. The best choice is to back up and restore using the same version of DOS. Do not ever mix versions of MS-DOS and PC DOS.

GUIDED ACTIVITY

USE THE PARTIAL RESTORE PROCESS: This activity is a follow-up to the preceding one. During this activity, you will restore several document (.DOC) files from Backup Disk 01 to the C:\WEST subdirectory using the RESTORE command.

1. This exercise is meant as a follow-up to the last Guided Activity. BACKUP Disk 01 should still be in drive A:.

2. You should still be logged on to drive C: with the default path being C:\WEST. If this is not the case, use the CD command to change to the C:\WEST subdirectory.

3. Type **DIR** and press <ENTER>. The West Student Data Disk files should be in this directory; if not, go back to the previous step, and complete it before proceeding.

4. Type **DEL *.*** and press <ENTER>.

5. The screen will display the message "Are you sure (Y/N)?" Type **Y** and then press <ENTER>. This command will delete all files in the C:\WEST subdirectory.

6. Type **DIR** and press <ENTER>.

The subdirectory C:\WEST should not include any files. The screen should only list the two directory listings.

7. Type **CLS** and press <ENTER>.

8. Type **RESTORE A: C:*.DOC** and press <ENTER>.

This form of the command will restore from the backup disks all files ending with the .DOC extension.

9. When prompted to "Insert backup diskette 01 in drive A:", strike any key; the disk should already be in the drive.

The monitor screen will appear as shown in Fig. 8-6. These four files are the only .DOC files on Backup Disk 01.

```
C:\WEST>RESTORE A: C:*.DOC

Insert backup diskette 01 in drive A:
Strike any key when ready


*** Files were backed up 05-03-1989 ***

*** Restoring files from drive A: ***
Diskette: 01
\WEST\DEGAS.DOC
\WEST\EAGLE.DOC
\WEST\OVERDUE.DOC
\WEST\PCAT.DOC

C:\WEST>
```

Fig. 8-6. Screen response when completing partial restore.

10. Type **DIR** and press <ENTER>.

Notice that the four files listed below are now in the C:\WEST subdirectory.

DEGAS.DOC
EAGLE.DOC
OVERDUE.DOC
PCAT.DOC

11. Type **CLS** and press <ENTER>.

12. Type **RESTORE A: C:*.TXT** and press <ENTER>.

13. When prompted, press any key.

Three more files, all with the .TXT extension, will be restored to the C:\WEST directory.

14. Type **DIR** and press <ENTER>. The directory now includes seven files, plus two directory entries.

15. Type **CLS** and press <ENTER>.

16. Type **TYPE TRUCKIN.TXT** and press <ENTER>.

The file has been successfully restored. Figure 8-7 displays the characters of this ASCII file.

```
C:\WEST>TYPE TRUCKIN.TXT
Who comprises the world's largest unsupervised
work force?  Think for a moment.  Would you guess
waitresses, college professors, gas station
attendants, or free-lance writers?  Guess again.
This distinction belongs to the lonely trucker--but
not for long.

In March 1988, Geostar Corporation of Washington,
D.C., launched a communications satellite that
enables shipping firms to track their truckers
using an on-board tracking device.  Shippers using
the system can maintain surveillance over their
fleet of trucks.  The Geostar system can pinpoint a
truck within a fraction of a mile and report to the
shipping firm as often as every fifteen minutes.
Hey, Good Buddy, Big Brother is watchin' ya!

C:\WEST>
```

Fig. 8-7. TYPE command display for the TRUCKIN.DOC file.

ANOTHER SAFETY COMMAND--ATTRIB

Microsoft Corporation added another safety feature with the arrival of PC DOS 3.x. It is the ATTRIB command. The **ATTRIB** command can alter either the access status of a file or its archive status. First we discuss the access status.

SETTING THE READ-ONLY BIT

When a file is created on disk, a directory entry is also created. The eleventh byte of each directory entry is the **attribute byte**. Each bit composing the eleventh byte has a special meaning. One of the bits indicates whether the file is marked for read-only or for both **read-and-write** operations. This bit can be altered--changed from a 0 to 1, or vice versa--by using the external DOS command ATTRIB. The command's syntax is

```
ATTRIB [+R][d:][path][filename.ext]
```

or

```
ATTRIB [-R][d:][path][filename.ext]
```

If you wish to write-protect (that is, make read-only) a file named JUNKMAIL.TXT in the root directory on drive C:, the attribute bit must be set to a value of 1. Therefore, the command takes the form of

```
ATTRIB +R  C:JUNKMAIL.TXT
```

After setting the attribute to read-only, we can determine that the process was successful by entering the command

```
ATTRIB C:JUNKMAIL.TXT
```

DOS will respond with the message

```
R       C:\JUNKMAIL.TXT
```

If the file's attribute were set to a read-and-write state, then the message would be

```
C:\JUNKMAIL.TXT
```

#10 The R appearing at the left margin indicates that the file is read-only or write-protected. The read-only attribute may be reset to 0, thus changing the file to a read-and-write state, by using the following command form:

```
ATTRIB -R C:JUNKMAIL.TXT
```

Global wildcard characters may be used with the ATTRIB command; thus, a group of files may be write-protected by a single command line.

It is important to realize that any file that is write-protected using the ATTRIB command may be accessed and viewed. However, no modifications may be made to the file without changing the attribute bit to 0--i.e., so the file is not write-protected. This means that the ATTRIB command is useful primarily with files saved for archival purposes. Files that are modified daily are not good candidates for the ATTRIB command. Another use of ATTRIB would be to write-protect files that

should not be altered by casual users, hackers, or computer viruses. Examples of this type of file would include COMMAND.COM, and .EXE or .COM files.

> **NOTE:** You will find that the EDLIN command will not work on a write-protected file. EDLIN is designed to edit or modify a file, so access is denied when the file's attribute is set to read-only. The TYPE command, on the other hand, is designed to view a file--not modify it. A read-only file can be viewed using the TYPE command.

The ATTRIB command would be useful in storing **boilerplate documents** for a word processor. If a department, company, or any enterprise finds that a basic response, with minor modifications, meets the needs of most correspondence they receive, then a series of standard paragraphs may be created and stored as a separate file. When needed, the file is copied into correspondence to clients. Once a file with the attribute set to read-only is copied, the duplicate copy is not write-protected. Modifications and additions may be made to the new file, yet the boilerplate file remains intact and write-protected.

The same process could be used with generic worksheets created by a spreadsheet program. A standard worksheet is created to store monthly telephone traffic in a department. The file, MASTPHON.WS, is then write-protected using the ATTRIB command. Each month a new copy of the master file is created by using the command

```
COPY MASTPHON.WS PHONEJUL.WS
```

Our example shows the command format for the month of July. Subsequent files created for subsequent months might be named PHONEAUG.WS, PHONESEP.WS, PHONEOCT.WS, etc. Because the new files are copied, they lack the read-only attribute of the MASTPHON.WS file.

An added feature of the ATTRIB command relates to the erasing of files. A file marked as read-only by the ATTRIB command cannot be erased using either the DEL or ERASE command. However, the FORMAT command will eliminate read-only files.

SETTING THE ARCHIVE BIT

The second use of the ATTRIB command is to set the archive bit indicating that a file has been backed up. Whenevér a file is backed up, the archive bit (one of the bits in the attribute byte) is set to indicate that the file has been backed up. Once you modify the file, the flag is reset to indicate that the file has not been archived (backed up). The ATTRIB command can be used to reset this flag. There may be occasions when you want to reset the archive bit, thus forcing the BACKUP command (using the /M parameter) to archive the file, even though it has not been modified.

The syntax of this version of the ATTRIB command is

```
ATTRIB [+A][d:][path][filename.ext]
```

or

```
ATTRIB [-A][d:][path][filename.ext]
```

To turn on the archive bit of a file called NOMODIFY.TXT, you would use the command

```
ATTRIB +A NOMODIFY.TXT
```

The result would be to enable the BACKUP command with the /M parameter to back up the NOMODIFY.TXT file, even if it has not been changed since the last backup.

The following command form would turn off the archive bit and cause the file not to be backed up when using the /M parameter:

```
ATTRIB -A NOMODIFY.TXT
```

#10 You can view the status of a file with regard to the archive (or read-only) status by entering the ATTRIB command followed by the filename. That is,

```
ATTRIB NOMODIFY.TXT
```

If the archive attribute is turned on, the screen will respond with the message

```
A       NOMODIFY.TXT
```

If the A does not appear, the archive bit is turned off. Global wildcard characters may also be used with this version of the ATTRIB command; thus, a group of files may have their archive bit turned on (or off) by a single command line.

THE ATTRIB /S PARAMETER

The ATTRIB command has one more parameter. The /S parameter causes the ATTRIB command to act upon files in the current directory and all of its subdirectories. For instance, if you need to turn the archive bit on for all files in the C:\DATA directory and any subdirectories under it, you would execute the command form

```
ATTRIB +A C:\DATA\*.*/S
```

To mark all files in the D:\WORDS directory and any of its subdirectories as read-only, execute the command

```
ATTRIB +R D:\WORDS\*.*/S
```

> **NOTE:** You can set both the archive and the access bit at the same time. The + or -R and + or -A can be combined, as shown in the following command line:
>
> ```
> ATTRIB +R +A C:\WORDS*.*
> ```

GUIDED ACTIVITY

MARK FILES READ-ONLY WITH THE ATTRIB COMMAND: In this Guided Activity, you will create a small file using EDLIN, mark that file as read-only, copy it several times, and then delete all but the file marked by the attribute byte as read-only. This will demonstrate that when a file marked as read-only is copied, the copy is not write-protected, and that a file marked as read-only cannot be deleted.

1. Insert a blank formatted disk in drive A:.
2. Be sure you are logged on to the drive and path (C:\DOS) containing your DOS command files.
3. Type **DIR A:** and press <ENTER>. There should be no files on the disk.
4. Type **EDLIN A:MASTER.WS** and press <ENTER>.
5. When the EDLIN prompt (i.e., asterisk) appears, type **I** and press <ENTER>.
6. Type **Let's pretend this is a master copy**
7. Press <ENTER>.
8. Type **of a budget worksheet**
9. Press <ENTER>.
10. Type **created by your favorite electronic**
11. Press <ENTER>.
12. Type **spreadsheet program.**
13. Press <ENTER>.

14. Press <CTRL-C>.

15. Type **L** and press <ENTER> to verify that the copy is correct.

Your screen should look like that shown in Fig. 8-8. If the text is incorrect, modify your file.

```
C:\WEST>DIR A:

 Volume in drive A has no label
 Directory of  A:\

File not found

C:\WEST>EDLIN A:MASTER.WS
New file
*I
       1:*Let's pretend this is a master copy
       2:*of a budget worksheet
       3:*created by your favorite electronic
       4:*spreadsheet program.
       5:*^C

*L
       1: Let's pretend this is a master copy
       2: of a budget worksheet
       3: created by your favorite electronic
       4: spreadsheet program.
*
```

Fig. 8-8. Text of the MASTER.WS file.

16. Type **E** and press <ENTER> to exit and save the file.

17. Type **ATTRIB +R A:MASTER.WS** and press <ENTER>.

The file is now write-protected and marked as read-only by the ATTRIB command. The next step will verify that MASTER.WS is write-protected.

18. Type **CLS** and press <ENTER>.

19. Type **ATTRIB A:*.*** and press <ENTER>.

```
C:\WEST>ATTRIB A:*.*
  A    R  A:\MASTER.WS

C:\WEST>
```

Fig. 8-9. Response to the ATTRIB A:*.* command.

The screen message should look like that shown in Fig. 8-9. The R to the left of the filename indicates that the attribute byte is set to read-only. Further, the archive bit (at the lefthand margin) is set on; that is, it indicates the file has been created or modified since the last backup. To reconfirm that MASTER.WS is protected from modification, try to edit it using EDLIN.

20. Type **EDLIN A:MASTER.WS** and press <ENTER>.

The screen will return with the message

File is READ-ONLY

The next few steps will create duplicates of the MASTER.WS file under four different filenames.

21. Type **COPY A:MASTER.WS A:JUNE89$$.WS** and press <ENTER>.

22. Press <F2>.

23. Type **N**

You will see that the function keys used for editing in EDLIN operate the same way at the DOS level. The <F2> Key followed by a keystroke repeats the template until the character entered (after pressing <F2>) is found. Characters stored in the keyboard buffer from the last line entered are referred to as the *template*.

24. Type **LY** and press <F3>. <F3> repeats the remainder of the template from the keyboard buffer.

25. Press <ENTER>.

26. Press <F2>.

27. Type **J**

28. Type **AUG**

29. Press the <DEL> Key to delete one character from the DOS template (the last command entered and still stored in the keyboard buffer).

30. Press <F3>. The line should now read "COPY A:MASTER.WS A:AUG89$$.WS."

31. Press <ENTER>.

32. Press <F2>.

33. Type **:**

34. Press <F2> again.

35. Type **:**

36. Type **:SEP**

37. Press <F3>. The line should now read "COPY A:MASTER.WS A:SEP89$$.WS."

38. Press <ENTER>.

Check to be sure that the newly created files are on the disk in drive A:.

39. Type **DIR A:** and press <ENTER>.

The following files should be on the disk:

MASTER.WS
JUNE89$$.WS
JULY89$$.WS
AUG89$$.WS
SEP89$$.WS

The MASTER.WS file is read-only, but what about the files copied from it?

40. Type **CLS** and press <ENTER>.

41. Type **EDLIN A:JUNE89$$.WS** and press <ENTER>.

42. When the "End of input file" message appears, type **L** and press <ENTER>.

The text file should appear as shown in Fig. 8-10.

```
C:\WEST>EDLIN A:JUNE89$$.WS
End of input file
*L
       1:*Let's pretend this is a master copy
       2: of a budget worksheet
       3: created by your favorite electronic
       4: spreadsheet program.
*
```

Fig. 8-10. Screen display when listing a file using EDLIN.

In the next few steps, you will do a slight modification on the file.

43. Type `1` (that is, a number one) and press <ENTER>.

44. Press <F2>, and type `m`

45. Type `copy`

46. Press <ENTER>.

47. Type `L` and press <ENTER>.

The text file should now appear as shown in Fig. 8-11. Notice that the word "master" is gone from line one.

```
C:\WEST>EDLIN A:JUNE89$$.WS
End of input file
*L
       1:*Let's pretend this is a master copy
       2: of a budget worksheet
       3: created by your favorite electronic
       4: spreadsheet program.
*1
       1:*Let's pretend this is a master copy
       1:*Let's pretend this is a copy
*L
       1:*Let's pretend this is a copy
       2: of a budget worksheet
       3: created by your favorite electronic
       4: spreadsheet program.
*
```

Fig. 8-11. Screen display of JUNE89$$.WS.

48. Type `E` and press <ENTER> to exit and save the file.

49. Type `DIR A:` and press <ENTER>.

Notice that the file JUNE89$$.WS is now only 113 bytes long rather than 120 bytes long (which is the length of MASTER.WS). In addition, a new file, automatically created by DOS and titled JUNE89$$.BAK, has been created. You were definitely able to modify the copy of a read-only file--even though the original was write-protected.

50. Type **ATTRIB A:*.*** and press <ENTER>.

The newly created files have the archive attribute bit set on (A is at the lefthand margin). These files would be backed up if you used the BACKUP command with the /M parameter.

Finally, the next few steps attempt to assure you that read-only files cannot be deleted! Only the MASTER.WS file is marked (using the ATTRIB command) as a read-only file.

51. Type **DEL A:*.WS** and press <ENTER>. This command will delete all files on drive A: that end with the .WS extension.

52. Type **DIR A:** and press <ENTER>.

Of the files created in this exercise, only two remain:

MASTER.WS
JUNE89$$.BAK

Clearly, the file marked as read-only withstood the DEL command and is still safely available to you. We did not instruct DOS to delete the other file with the .BAK extension.

COPYING GROUPS OF FILES WITH XCOPY

The BACKUP and RESTORE commands have several distinct disadvantages.

1. You cannot access or modify files backed up with the BACKUP command until they are restored.
2. Though you can restore backed-up files to a different drive, they must be restored to a directory having the same name as the source directory. That is, if you use BACKUP to back up files from the C:\WORD\BOOK subdirectory, the files must be restored to a \WORD\BOOK directory (the drive may be different, but not the path).
3. The track record is poor when restoring a group of files unless you use the very same version of DOS that was originally used to back up the files.
4. The BACKUP command copies files one at a time. The first file must be read into memory from the source disk and then copied to the target disk. This process is slow because each file must be found, read, and copied before the next file is found, read, and copied to the target disk.

The XCOPY command provides an alternative to each of these problems. It can also be used to back up files. It cures the above four shortcomings of the BACKUP command, but it has one outstanding weakness itself. We will discuss the weakness first.

XCOPY can only copy the number of files that may be stored on one diskette at a time. Suppose you use the XCOPY command with a global wildcard that would select 400K of files; however, your system only uses 360K diskettes. XCOPY will begin copying until the target disk is full and then abort the copying process--just as the COPY command would do. This also means you cannot use XCOPY to back up or make a duplicate of a file that exceeds the size of your target disk. In such a case, you must use the BACKUP command.

In the four areas listed above, XCOPY wins "hands down" because of the following:

1. Files copied with XCOPY may be read and modified immediately.
2. XCOPY can copy from any directory on one disk to any directory on another disk. The target and source directories do not need to be identical.
3. XCOPY has only been available since DOS version 3.2, but files copied using 3.2 are readable by any other version of DOS.
4. The final advantage is speed. XCOPY copies as many files into memory as RAM will hold and then begins writing those files out to the target disk. With fewer reads of the source disk and fewer writes to the target diskette, the copying speed is greatly enhanced.

There is one other minor annoyance with XCOPY--you cannot use this command to copy hidden, system, or read-only files. The XCOPY command is an external command that uses the following syntax:

#12

```
[d:][path]XCOPY s:[path][filename.ext]
    t:[path][filename.ext]
```

To copy all files from the current directory on drive A: to drive B:, the command form would be

```
XCOPY A: B:
```

You can use filenames and global wildcard characters to restrict the files to be copied. The XCOPY command also has several parameters available.

/A	Only copies files with the archive bit set on, but does not change the source file archive bit.
/D:mm-dd-yy	Only copies to the target disk any files created on or after the date specified.
/E	Copies or creates subdirectories on the target disk even if they end up being empty. Without the /E parameter, empty directories on the source disk are not copied or created on the target disk.

/M	Only copies files with the archive bit set on (1) and then turns the source file archive bit off (0).
/P	Prompts you with a (Y/N?) before copying each file.
/S	Copies all files from the current directory and its subdirectories. If this parameter is omitted, XCOPY only transfers files from the current (or specified) directory on the source disk to the target disk.
/V	Verifies that sectors written from the source to the target disk are written correctly.
/W	Causes DOS to wait until you insert a disk in the source drive before checking for source files.

GUIDED ACTIVITY

USE THE XCOPY COMMAND: In this Guided Activity, you will use the XCOPY command to transfer all files created in the last Guided Activity to a subdirectory under the \WEST directory on your hard disk.

1. Insert the disk used in the last Guided Activity in drive A:. It should include two files: MASTER.WS and JUNE89$$.BAK.

2. Be sure you are logged on to the hard-disk directory named \WEST.

3. Type **DIR** and press <ENTER>. This directory should have seven files and two directory listings.

4. Type **CLS** and press <ENTER>.

5. Type **MD WKSHEETS** and press <ENTER>.

6. Type **CD WKSHEETS** and press <ENTER> to change to the new directory.

7. Type **XCOPY A:*.* C:** and press <ENTER>.

If you were to use the COPY command, DOS would have to read the first file from drive A:, copy it to drive C:, read the second file from A:, and copy again to drive C:. If more than two files were on the source disk, the process would continue until all files were copied--one at a time.

On the other hand, when the XCOPY command is executed, DOS first reads the command into memory (it is an external DOS command). Next, XCOPY reads as many files from the source drive into memory as it can hold in RAM and then transfers the files to the target drive. Only if the size of the source files exceeds the amount of available RAM does XCOPY return to the source drive to read again.

8. Type **DIR** and press <ENTER>.

Figure 8-12 displays the screen messages generated by executing the XCOPY command. Once XCOPY is executed, the "Reading source file(s)" message is displayed. Both files are transferred in one "read" from drive C:. Figure 8-12 also includes the results of the DIR command.

```
C:\WEST>MD WKSHEETS

C:\WEST>CD WKSHEETS

C:\WEST\WKSHEETS>XCOPY A:*.* C:

Reading source file(s)...
A:MASTER.WS
A:JUNE89$$.BAK
        2 File(s) copied

C:\WEST\WKSHEETS>DIR

 Volume in drive C is JONATHAN'S
 Directory of  C:\WEST\WKSHEETS

.            <DIR>       5-04-89   6:18a
..           <DIR>       5-04-89   6:18a
MASTER   WS         120   5-03-89   8:54p
JUNE89$$ BAK        120   5-03-89   8:54p
        4 File(s)   5646336 bytes free

C:\WEST\WKSHEETS>
```

Fig. 8-12. Screen messages from the XCOPY command after transferring files to drive C:.

In the next few steps, you will delete the files from drive A: and then use the XCOPY command (with the /S parameter) to copy files from the \WEST and \WEST\WKSHEETS directories to the A: drive. This demonstrates that XCOPY can back up more than one directory at a time.

9. Type **DEL A:*.*** and press <ENTER>.

10. When prompted "Are you sure (Y/N)?", type **Y** and press <ENTER>.

11. Type **DIR A:** and press <ENTER>. The disk in drive A: still contains the MASTER.WS file which has its attribute byte set to read-only. The next step removes the protection, allowing you to delete the write-protected file.

12. Type **ATTRIB -R A:*.*** and press <ENTER>.

13. Type **DEL A:*.*** and press <ENTER>.

14. When prompted "Are you sure (Y/N)?", type **Y** and press <ENTER>.

15. Type **DIR A:** and press <ENTER>. You will see a message stating "File not found."

16. Type **CD\WEST** and press <ENTER>. Your default drive and path should now be C:\WEST.

17. Type **CLS** and press <ENTER>.

18. Type **DIR** and press <ENTER>.

Only seven files should be in this directory. Figure 8-13 lists the files, two dot subdirectories (.<DIR> and ..<DIR>), and one regular subdirectory named WKSHEETS.

```
C:\WEST>DIR

 Volume in drive C is JONATHAN'S
 Directory of  C:\WEST

.            <DIR>       4-30-89   5:04p
..           <DIR>       4-30-89   5:04p
DEGAS    DOC     4268    5-02-89   7:41p
EAGLE    DOC     1056    5-02-89   7:45p
OVERDUE  DOC      617    3-04-89  10:17a
PCAT     DOC      679    4-22-89   7:41p
TOUCHY   TXT      460    5-02-89   7:18p
TECHNO   TXT     2112    5-02-89   7:00p
TRUCKIN  TXT      713    5-02-89   7:12p
WKSHEETS     <DIR>       5-04-89   6:18a
       10 File(s)   5644288 bytes free

C:\WEST>
```

Fig. 8-13. DIR listing of files in C:\WEST directory.

19. Type **CLS** and press <ENTER>.

20. Type **XCOPY *.* A:/S**.

Before you press <ENTER>, be prepared to watch the drive C: and A: lights. First DOS reads the external XCOPY command into memory. Then, after reading the files on drive C: into memory, the C: drive light does not show activity again. If you use the COPY command, DOS reads the first file from drive C:, copies it to drive A:, reads the next file from drive C:, and copies again to drive A:. This process continues until all files on the source disk are copied, one by one, to the target disk.

The more efficient XCOPY command reads as many files from drive C: into memory as it can hold in RAM, and then it copies them to drive A:. Once you press <ENTER>, the XCOPY command is read into memory; then DOS accesses drive C: again to read all files into memory and does not return to drive C: until the process is complete. Watch the drive lights.

21. Press <ENTER>.

Figure 8-14 displays the screen activity from executing the XCOPY command.

```
C:\WEST>XCOPY *.* A:/S

Reading source file(s)...
DEGAS.DOC
EAGLE.DOC
OVERDUE.DOC
PCAT.DOC
TOUCHY.TXT
TECHNO.TXT
TRUCKIN.TXT
WKSHEETS\MASTER.WS
WKSHEETS\JUNE89$$.BAK
        9 File(s) copied

C:\WEST>
```

Fig. 8-14. XCOPY screen activity.

22. Type **DIR A:** and press <ENTER>.

The disk in drive A: contains the seven files from the West Student Data Disk plus a new subdirectory that was not originally on drive A:. This new directory, A:\WKSHEETS, is drawn from the directory of the same name on drive C:.

Let's check the C:\WEST\WKSHEETS subdirectory to be sure the two files on drive C: are properly copied to the target disk in drive A:.

23. Type **A:** and press <ENTER> to change the default drive to A:.

24. Type **CD WKSHEETS** and press <ENTER>. Your default directory is now A:\WKSHEETS.

25. Type **DIR** and press <ENTER>.

The two files, MASTER.WS and JUNE89$$.BAK, are in fact in the subdirectory.

The next two steps return you to the root directory on drive A: and then to the \WEST directory on drive C:.

26. Type **CD** and press <ENTER>.

27. Type **C:** and press <ENTER>.

THE REPLACE COMMAND--A HANDY OPTION

When using the XCOPY command to back up important files, the REPLACE command is a handy companion. REPLACE is an external DOS command. The syntax of the command is

```
[d:][path]REPLACE [s:][path]filename.ext [t:]
```

If you had used the XCOPY command to back up chapters of a manuscript (titled UNIT1.DOC, UNIT2.DOC, UNIT3.DOC, and UNIT4.DOC), you could periodically replace the backed-up versions with the newer versions (presumably on your hard disk) by entering the command

```
REPLACE UNIT?.DOC A:
```

This command form assumes the following:

1. You are logged on to the hard-disk directory that contains the four unit files.
2. Only four files (UNIT1.DOC through UNIT4.DOC) exist on the target and source disk.
3. You wish to update the backup disk in drive A:.

If UNIT3.DOC exists on the default drive but not on drive A:, it will not be copied to the disk in drive A:. The REPLACE command does just as its name implies--without a special parameter, it replaces files. If a file does not exist on the target disk, it does not copy it.

There are several REPLACE command parameters that provide greater flexibility. They are:

/A Copies all files from the source to the target disk that do not already exist on the target drive.
/P Prompts you to respond (Y/N) to each proposed replacement. If you type **Y**, the file is copied to the target. If you type **N**, it is not copied.
/R Replaces files that are marked read-only on the target disk.
/S Searches all directories and subdirectories of the target for the specified file(s).
/W Waits until you insert a disk in the source drive before proceeding.

Most of these parameters may be combined. However, the /A parameter may not be used with the /S parameter. If you combine the /A and /W parameters, the REPLACE command copies all files from the source disk to the target disk (if they do not already exist on the target) and prompts you to "Press any key to begin adding file(s)." If you use the /W parameter alone, the prompt is slightly different. It asks you to "Press any key to begin replacing file(s)."

In the earlier example with the four unit files, suppose the UNIT3.DOC file did not exist on the target drive but did on the source drive. It could be copied to the target by using the following command line.

```
REPLACE UNIT?.DOC A:/A
```

REPLACE COMMAND ERROR CODES

DOS returns an error code (that can be determined using the IF ERRORLEVEL batch file command) if the REPLACE command is unsuccessful. The most common error codes are listed below.

2 No source files were found matching the parameters of the command.
3 Either the source or target path cannot be found.
5 Access to copy the file was denied. Most likely the file is marked read-only. To copy this type of file, use the /R parameter.
8 Insufficient memory is available to load and execute the REPLACE command.
11 The form of the REPLACE command or the number of parameters is invalid.
15 An invalid drive name was specified for either the source or the target drive.
22 DOS does not recognize the command, which indicates that an incorrect version of DOS is being used.

GUIDED ACTIVITY

USE THE REPLACE COMMAND: In this Guided Activity, you will modify a small file using EDLIN and then use the REPLACE command to update the file on a backup disk.

1. Be sure the data disk from the last Guided Activity is still in drive A:.

2. Log on to the \WEST directory on your hard disk.

3. Type **ATTRIB *.*** and press <ENTER> to display the attribute byte settings for all files on the current path and directory.

Figure 8-15 shows the results of this command. All seven files have the archive bit set off; that is because they are restored files and DOS assumes a backup exists--and the files have not been modified since their restoration.

```
C:\WEST>ATTRIB *.*
          C:\WEST\DEGAS.DOC
          C:\WEST\EAGLE.DOC
          C:\WEST\OVERDUE.DOC
          C:\WEST\PCAT.DOC
          C:\WEST\TOUCHY.TXT
          C:\WEST\TECHNO.TXT
          C:\WEST\TRUCKIN.TXT

C:\WEST>
```

Fig. 8-15. ATTRIB command.

4. Type **CLS** and press <ENTER>.

5. Type **REPLACE *.* A:** and press <ENTER>. The command causes all seven files to be replaced on drive A: with the versions from drive C:.

Figure 8-16 displays the results of the REPLACE command.

```
C:\WEST>REPLACE *.* A:
 Replacing A:\DEGAS.DOC
 Replacing A:\EAGLE.DOC
 Replacing A:\OVERDUE.DOC
 Replacing A:\PCAT.DOC
 Replacing A:\TOUCHY.TXT
 Replacing A:\TECHNO.TXT
 Replacing A:\TRUCKIN.TXT
7 file(s) replaced
C:\WEST>
```

Fig. 8-16. REPLACE command.

6. Type **CLS** and press <ENTER>.

In the next few steps, you will create a new file and then use a new parameter with the REPLACE command. The /A parameter appends files from the source disk onto the target disk--if they do not already exist on the target disk.

7. Type **REPLACE *.* A:/A** and press <ENTER>.

You will see a "No files added" message on the screen. This form of the REPLACE command only adds a file to the target disk if the file does not already exist on the target disk.

8. Type **CLS** and press <ENTER>.

9. Type **EDLIN EMPTY.ONE** and press <ENTER>.

10. At the EDLIN prompt, type **I** and press <ENTER>.

11. Type `This is an empty, useless file.`

12. Press <ENTER> five times.

13. Press <CTRL-C>.

14. Type `L` and press <ENTER>. Your screen should look like the one shown in Fig. 8-17.

```
C:\WEST>EDLIN EMPTY.ONE
New file
*I
       1:*This is an empty, useless file.
       2:*
       3:*
       4:*
       5:*
       6:*^C

*L
       1: This is an empty, useless file.
       2:
       3:
       4:
       5:
*
```

Fig. 8-17. File created using EDLIN in the C:\WEST directory.

15. If your file matches the file in Fig. 8-17, type `E` and press <ENTER>.

16. Type `CLS` and press <ENTER>.

17. Type `REPLACE *.* A:/A` and press <ENTER>.

The screen will display the messages shown in Fig. 8-18. This form of the REPLACE command added one file from the source drive (C:\WEST) to the target disk in drive A:.

```
C:\WEST>REPLACE *.* A:/A

 Adding A:\EMPTY.ONE

1 file(s) added

C:\WEST>
```

Fig. 8-18. Screen messages generated by the REPLACE /A command.

18. Type **DIR A:** and press <ENTER>.

The EMPTY.ONE file now appears on the directory listing for drive A:. See Fig. 8-19.

```
C:\WEST>REPLACE *.* A:/A

 Adding A:\EMPTY.ONE

1 file(s) added

C:\WEST>DIR A:

 Volume in drive A has no label
 Directory of  A:\

DEGAS    DOC      4268   5-02-89   7:41p
EAGLE    DOC      1056   5-02-89   7:45p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
TOUCHY   TXT       460   5-02-89   7:18p
TECHNO   TXT      2112   5-02-89   7:00p
TRUCKIN  TXT       713   5-02-89   7:12p
WKSHEETS      <DIR>      5-06-89   7:11a
EMPTY    ONE        42   5-06-89   7:25a
        9 File(s)    344064 bytes free

C:\WEST>
```

Fig. 8-19. EMPTY.ONE file appears in DIR listing.

19. Type **CLS** and press <ENTER>.

20. Type **ATTRIB +R A:EMPTY.ONE** and press <ENTER>. This step write-protects the EMPTY.ONE file in drive A: only.

21. Type **ATTRIB A:*.*** and press <ENTER>. Only EMPTY.ONE should be marked as read-only. All the files on drive A: have the archive bit set on.

22. Type **CLS** and press <ENTER>.

23. Type **REPLACE *.* A:/P** and press <ENTER>. This command form causes the REPLACE operation to prompt you whether to copy each file meeting the command specifications (in this case, *.* means all files).

24. The first message will read "Replace A:\DEGAS.DOC (Y/N)?" Each time you receive a prompt, type **Y** and press <ENTER> to instruct DOS to copy the listed file.

After you have typed **Y** for each prompt, REPLACE finally completes copying each file except the last one. You will see an error message, as shown in Fig. 8-20. Access is denied when you try to overwrite a read-only file. By adding the /R parameter to the REPLACE command, a file marked read-only can be overwritten.

```
C:\WEST>REPLACE *.* A:/P
 Replace A:\DEGAS.DOC? (Y/N) Y
 Replacing A:\DEGAS.DOC
 Replace A:\EAGLE.DOC? (Y/N) Y
 Replacing A:\EAGLE.DOC
 Replace A:\OVERDUE.DOC? (Y/N) Y
 Replacing A:\OVERDUE.DOC
 Replace A:\PCAT.DOC? (Y/N) Y
 Replacing A:\PCAT.DOC
 Replace A:\TOUCHY.TXT? (Y/N) Y
 Replacing A:\TOUCHY.TXT
 Replace A:\TECHNO.TXT? (Y/N) Y
 Replacing A:\TECHNO.TXT
 Replace A:\TRUCKIN.TXT? (Y/N) Y
 Replacing A:\TRUCKIN.TXT
 Replace A:\EMPTY.ONE? (Y/N) Y
 Replacing A:\EMPTY.ONE
Access denied  A:\EMPTY.ONE

7 file(s) replaced

C:\WEST>
```

Fig. 8-20. REPLACE command prompts with the error message.

25. Type **CLS** and press <ENTER>.

26. Type **REPLACE EMPTY.ONE A:/R** and press <ENTER>.

This time, because the /R parameter was added to the REPLACE command, the replacement is successful, as shown by Fig. 8-21.

```
C:\WEST>REPLACE EMPTY.ONE A:/R

 Replacing A:\EMPTY.ONE

1 file(s) replaced

C:\WEST>
```

Fig. 8-21. Replacement of MASTER.WS by the REPLACE command.

RECOVERING THE GOOD "PARTS" OF A FILE

The RECOVER command is helpful in two situations in which data might otherwise be lost and you need to retrieve the data.

1. When a sector of a file goes bad, RECOVER can be used to retrieve data in the good portions of a file; that is, in the sectors that are not damaged. In this case, RECOVER is used on a single file. RECOVER ignores the bad sectors and recreates as much of the file as is contained in readable sectors.

2. On the other hand, if the directory table of a disk is damaged, the RECOVER command is used to pull together all readable portions of a disk full of files.

RECOVER is an external command. The syntax of the command is

```
[d:][path]RECOVER t:[path][filename.ext]
```

If you wished to recover all files on a disk in drive A:, the command form would be

```
RECOVER A:
```

This form assumes that you have either invoked a PATH command to the DOS directory where the RECOVER command is stored or logged on to the DOS subdirectory.

If you wish to reconstruct a single file named TRASHED.DOC on the disk in drive B:, use the following command form:

```
RECOVER B:TRASHED.DOC
```

RECOVER can be dangerous if you fail to specify the damaged file and only indicate the target drive. The command has no way of knowing whether the data in a directory is valid or not. If you fail to specify a file, you may do damage to the entire disk as RECOVER attempts to recover good and bad files alike. Where possible, create a backup of the disk or file before using the RECOVER command.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`ATTRIB +A Filename`	Turns the specified file's archive bit on.
`ATTRIB -A Filename`	Turns the specified file's archive bit off.
`ATTRIB +R Filename`	Marks the specified file as read-only.
`ATTRIB -R Filename`	Marks the specified file as read-and-write accessible.
`BACKUP C:\*.* A:/S`	Backs up all files from all directories and subdirectories on hard-disk C: to floppy disks in the A: drive.
`BACKUP C:*.* A:/M`	Backs up all modified files (since the last backup) in the current directory of hard-disk C: to floppy disks in the A: drive.
`RECOVER A:`	Recovers data from any good sectors on the disk in drive A:.
`RECOVER B:MESSEDUP.DOC`	Recovers data from any good sectors storing portions of the file MESSEDUP.DOC on drive B:.
`REPLACE *.* A:/P`	Copies all files from the default drive and path to drive A after prompting you. If you respond with Y(es), the file is copied. A response of N(o) causes the file not to be copied.
`REPLACE *.* A:/R`	Copies read-only files from the default drive and path to drive A:.
`REPLACE A:*.* B:`	Copies all files from A: to B: if the files already exist on B:.
`REPLACE A:*.* B:/A`	Copies all files from A: to B: if the files do not already exist on B:.
`RESTORE A: C:\*.*/S`	Restores all files from the backup diskettes in drive A: to the appropriate directory or subdirectory on hard-disk C:.

`RESTORE A: C:Filename`	Restores the specified file from the backup disk in drive A: to the hard-disk C:.
`XCOPY A:*.DOC B:`	Copies all document files with the .DOC extension from A: to B:.
`XCOPY A: B:`	Copies all files from drive A: to B:.

REVIEW QUESTIONS

1. How are the uses of the COPY and DISKCOPY commands different from the uses of the BACKUP and RESTORE commands?

2. Name at least three concerns you should be aware of when using the BACKUP and RESTORE commands.

3. Which DOS commands should never be in effect when using the BACKUP or RESTORE command?

4. What are the two primary methods used for copy-protecting application software? What impact does each have on the backup and restore process?

5. What is a disadvantage of the DOS BACKUP and RESTORE commands in comparison to tape backup systems?

6. What two purposes does the ATTRIB command serve?

7. Why would globally changing all files to read-only not be wise?

8. Give an example in which it would be useful to write-protect a file with the ATTRIB command.

9. Can the REPLACE command replace a file that has been write-protected using the ATTRIB command?

10. How would you find out the archive status and the read-only status of a file?

11. If you select (using wildcard characters) more than 2MB of files, what happens when you try to execute the XCOPY command (assuming the target is a diskette drive)?

12. Which three types of files cannot be copied using the XCOPY command?

13. What is the effect of using the /A parameter with the REPLACE command?

14. What is the effect of not using any parameters with the REPLACE command?

15. Why would you use the RECOVER command?

UNIT EXERCISES

1. Create a batch file to change the archive bit so that the file selected (use a replaceable parameter) will be backed up the next time you use the BACKUP/M command--even if the file has not been modified. Call the batch file ARCIT.BAT. ATTRIB +[A][B] FILENAME.EXT

2. Create two batch files. One called READONLY.BAT marks a file as read-only. The second batch file called READWRIT.BAT should remove the read-only status from a specified file. In both cases, use a replaceable parameter.

DOCUMENTATION RESEARCH

To answer the following questions, refer to pages 7-25 through 7-30, 7-180 thru 7-182, and 7-189 through 7-192 in the *IBM Disk Operating System Version 3.30 Reference* manual. If using *Microsoft MS-DOS User's Reference*, review pages 35 through 37, 103, and 108 through 109 to answer the questions.

1. What will happen if you are sharing files on a network and you attempt to back up files to which you do not have access privileges?*

2. What rule applies to the target diskette and target drive when using the /F parameter with the BACKUP command?*

3. How is the date format established for use with the BACKUP /D command?*

4. Does the BACKUP command ever erase files? Explain your answer.

5. Should the source diskette ever be write-protected when using the BACKUP command? Explain your answer. *

6. How is the DOS 3.30 BACKUP command different from the DOS 3.2 BACKUP command?*

7. Can the RECOVER command be used on a network disk?

8. If you attempt to recover all files from a disk that has more than 112 files, what special action must be taken?*

9. Which DOS files cannot be restored using the RESTORE command? If the files cannot be restored, how would you get these files back on your disk?

10. Which three RESTORE command parameters cannot be used together?*

* Not covered in the MS-DOS documentation.

APPLICATION F

PART 1: RESTORING AN ASCII TEXT FILE

In Unit 8, we discussed the DOS commands used to back up and restore program and data files. In this application, you will restore a file that was created earlier with the DOS BACKUP command.

A. Obtain a copy of two files from your instructor. They are called BACKUP.001 and CONTROL.001. These files were created with the BACKUP command when a single file was backed up.

B. Restore the original file using the two backup files.

C. Once restored, redirect the original file to a printer.

D. Turn in a printed copy of the original file to your instructor.

PART 2: SAFEGUARDING FILES WITH THE ATTRIB COMMAND

In this application, you will create a small ASCII text file using EDLIN and then mark it as a read-only file.

A. Create a text file using EDLIN. Give the text your first name, up to eight characters, as a filename. Use the following text as the contents of the file. Each line should be on a different line in your file (created with EDLIN), as shown

I am a firm believer in procrastination.
Last year I even thought about forming a club of
procrastinators--but never got around to it.
I'm not sure it was a wise idea anyway.
I was going to talk to some friends about serving as
officers--but figured I could do it tomorrow.
Finding officers might be difficult though.
What true procrastinator would want to start
a meeting on time?
In fact, who would chair the meetings?
We'd never get around to elections.
Can you imagine being the treasurer?
Everyone would defer paying dues until December 31st.
Committees would never meet--until tomorrow.
Our creed could be:
"Never do today what you can postpone until tomorrow."
Most of our members would prefer to do nine stitches
rather than a single one in time.
Well, maybe it is just an idea "before its time."
I guess I'll think it over some more--tomorrow.

B. Use the appropriate DOS command to mark the file as read-only.

C. Double-check your work. Try to delete the file. If it cannot be deleted, you have correctly given the file the read-only attribute.

D. Turn in the file to your instructor on a diskette.

PART IV

ADDITIONAL DOS COMMANDS

The final units of this text cover several additional, crucial DOS concepts. Primary among these concepts are programming tools that include pipes, filters, redirection of input and output, background printing, and creating virtual disks.

The text then briefly examines DOS as it relates to networking microcomputers and file sharing. The text concludes with an examination of multitasking, multiprocessing, and future directions for DOS development.

UNIT 9

DOS REDIRECTION, PIPES, AND FILTERS

OBJECTIVES

After completing this unit, you will be able to

1. list and define standard PC DOS devices;
2. list and define the default DOS standard devices;
3. explain the difference between a serial port and a parallel port;
4. explain one use of the NUL: device;
5. define DOS redirection and give an example using DOS commands;
6. redirect output to a new disk file;
7. append output to an existing disk file by using redirection;
8. explain the difference between pipes and filters;
9. list and use the DOS filters;
10. explain the parameters that may be used with the SORT filter;
11. explain the parameters that may be used with the FIND filter;
12. explain how the MORE filter is used;
13. discuss the ASCII collating sequence and how it affects the SORT filter.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. FIND
2. MORE
3. SORT.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): baud rate, parallel port, secondary storage, serial port, and standard devices.
____Guided Activity: Redirect DOS Input and Output.
____Guided Activity: Use the SORT and MORE Filters.
____Guided Activity: Use the FIND Filter.
____Review Questions.
____Unit Exercises.
____Documentation Research.

Place a check in front of those assigned by your instructor.

STANDARD PC DOS DEVICES

Despite covering a great deal of territory relative to DOS commands, we have yet to explore the area of **standard devices**. PC DOS permits us that luxury because it assumes that all activities use the two default devices, one input and the other output--until we instruct the machine otherwise.

The default device for normal input operations is referred to as the console, specifically the console keyboard. The default output device is also the console, specifically the monitor screen. The console may also be referred to as a terminal or workstation. Thus, the operating system looks to the console for input and output unless specified otherwise. As already implied, the keyboard and the display monitor compose the console. In DOS shorthand, the form of the parameter is simply CON:. This parameter should look familiar because we used it in the COPY CON: command exercise.

As we consider additional standard devices, we must return to the basics of computing. What are standard input devices? What are standard output devices? The answer would include, in addition to the console, such items as a light pen, printer, modem, and disk drives. DOS is less concerned with the specific peripheral than it is with the method of interface. For DOS to accept input and send output to a peripheral, it makes available several standard devices known as **serial ports** or **parallel ports**. Serial ports are used to connect peripherals to your computer or to connect one PC to another. Parallel ports are most commonly used to connect a printer to your PC.

A parallel communications port on a PC is used to transmit eight bits over eight pins on the parallel cable simultaneously (i.e., one byte at a time). Keep in mind that one byte equals one character, such as a number, letter, or special character. The transfer of the eight bits must be handled concurrently so the eight bits represent the same character on the receiving end as they did on the transmitting end. Transmitted signals weaken over time and space and can get out of

synchronization. For this reason, parallel printers are placed close to the computers they serve. Parallel cabling should be no more than ten to fifteen feet in length, or you run the risk of distorting the data transmitted.

A serial port has the capability of transferring data one bit at a time over a cable. Thus, it requires eight transmissions to equal a single character (byte). Though transmission speeds are slower than for short-range parallel transmissions, serial cabling can be run greater distances without encountering the problems faced by parallel communications.

With serial transmissions, the **baud rate** must be specified and whether the parity is none, even, or odd. Serial printers must be synchronized to the same speed (baud rate) that the serial port is transmitting. Serial printers will include a table in their documentation indicating the proper attributes required for communications with your PC. Serial ports are also commonly used for data communications from one PC to another using a modem. The most common form of serial transmission is RS232.

The discussion of serial versus parallel communications is really a discussion of cabling requirements. Figure 9-1 shows the back of an IBM PC.

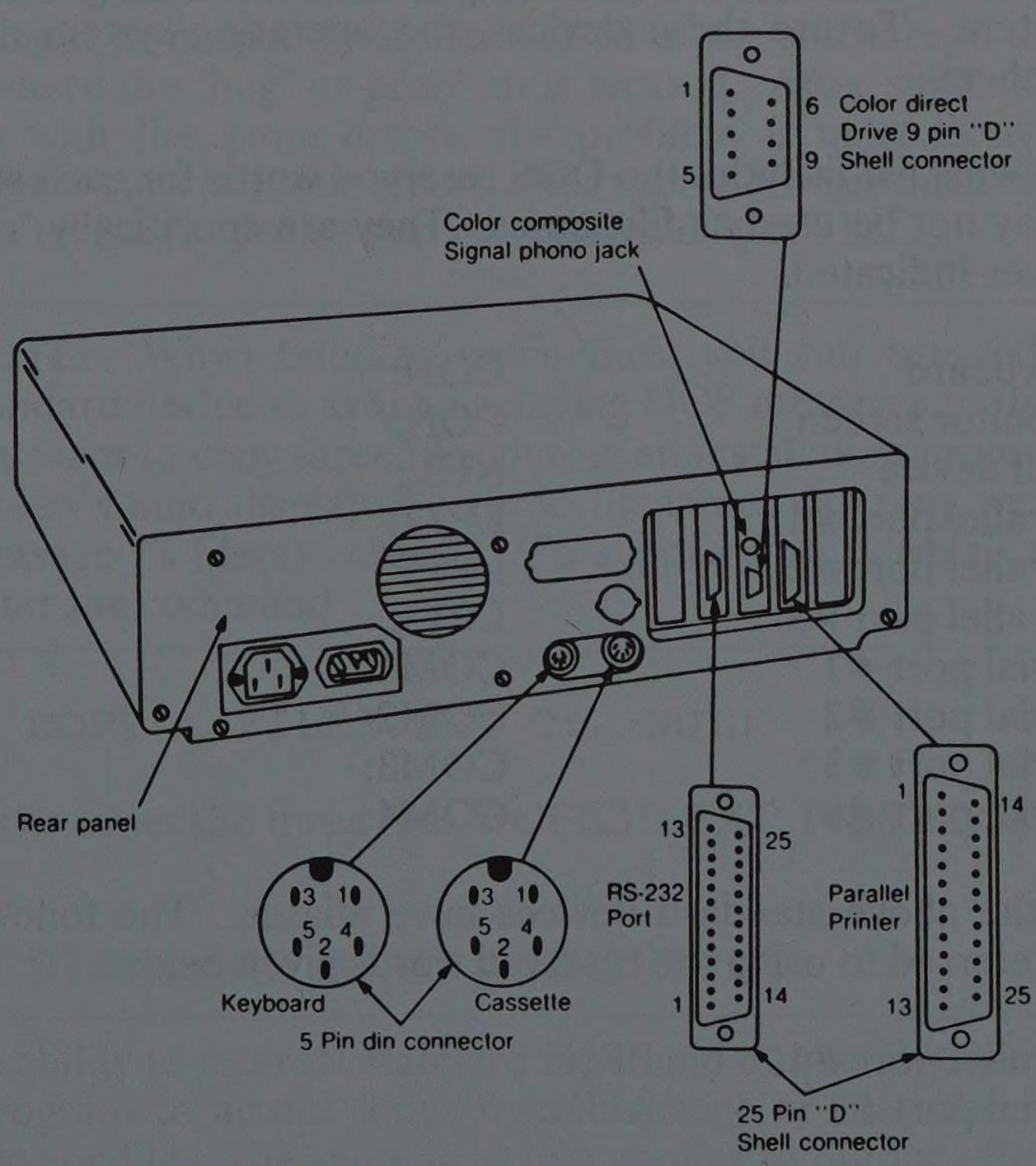


Fig. 9-1. Serial and parallel ports on an IBM PC.

The figure shows five types of cables used to connect your PC to its peripherals. The 25-pin parallel connector and the 25-pin serial connector are identified in the lower right corner of the illustration. The newer PS/2 models and some PC AT compatibles now use a nine-pin serial port adapter similar to the nine-pin port identified in the upper right corner of the figure (though that port is used for connection to the computer's monitor).

Returning to the discussion of standard devices, the following table indicates which type of port the listed peripherals use:

Device	Port
Light pen	Serial
Modem	Serial
Mouse	Serial or parallel
Network printer	Serial or parallel
Printer	Serial or parallel
Tape backup unit	Serial

Until the introduction of DOS 3.3, the IBM PC family of personal computers was structured to handle only two serial and three parallel ports. DOS 3.3 increased the number of devices by adding the capability of including two more serial ports on each system. To use these devices, the operating system needs to know how to address them.

The following list includes the DOS reserved words for each standard device. These terms may not be used as filenames. They are specifically "reserved" as names for the devices indicated.

• Keyboard	CON:
• Monitor screen	CON:
Null device	NUL:
Parallel port #1	LPT1:
Parallel port #2	LPT2:
Parallel port #3	LPT3:
Serial port #1	COM1:
Serial port #2	COM2:
Serial port #3	COM3:
Serial port #4	COM4:

Two of the above standard devices have aliases. The following standard devices may be referred to using the reserved words given below

Parallel port #1	PRN:
Serial port #1	AUX:

It may be easier to remember the reserved words if you know the root words from which they are derived:

AUX:	Auxiliary device
COM:	Communications port
CON:	Console
LPT:	Line printer
PRN:	Printer

Even though we have referred to each device name as ending with a colon, the colon is optional. For example, line printer #1 may be called PRN, PRN:, LPT1, or LPT1:.

Each of the reserved words representing a device may substitute for a parameter in a DOS command (where a filename might otherwise be used). We'll explore several examples in which this is the case.

A note on the NUL: standard device may be helpful. It is useful as a programmer's tool and can be used to substitute temporarily for an input or output device in the early stages of creating software. Further, NUL: can assist in the diagnosis of hardware and software problems. When a problem is encountered, the first step is to determine if the error is hardware- or software-related. By substituting NUL: for the input or output device stipulated in the software, a determination can be made on where the "bug" or problem is located. After substituting NUL:, if the program runs with the same errors, the problem is most likely software-related. If the program runs without errors after the NUL: substitution, the problem is with the input or output device that was replaced by NUL:.

NOTE: When building batch files, you can use redirection with the NUL: standard device to avoid displaying DOS messages to the screen. For instance, if you were to copy three files during an installation program (INSTALL.BAT), the screen would display the three filenames (as they were copied) and then the message "3 file(s) copied". To suppress any display to the screen, you could enter the command

```
COPY A:FILE?.TXT C: >NUL:
```

This assumes the three files are FILE1.TXT, FILE2.TXT, and FILE3.TXT.

REDIRECTION OF STANDARD INPUT AND OUTPUT

The ability to redirect standard input and output (I/O) is built into DOS. It enables a program to accept input from a device or file other than the standard default device or file. The same principle applies to output. Earlier, we indicated that the console (CON:) is the default I/O device for DOS. If the default devices were always used, then all input for any program would be restricted to the keyboard. Further, only the monitor screen could receive output. Such a situation would limit

the value of the computer because output could not be directed to modems or printers. Further, input could not be accepted from light pens, modems, graphic tablets, or any other peripheral. Redirection of standard input and output devices eliminates this restriction.

A specific example might help. The DIR command lists to the screen a directory of all files on a specified diskette. For example, with the West Student Data Disk in drive A:, the `DIR A:` command would list the following files:

```
TOUCHY.TXT
TECHNO.TXT
TRUCKIN.TXT
DEGAS.BDC
DEGAS.DOC
EAGLE.BDC
EAGLE.DOC
OVERDUE.DOC
PCAT.DOC
```

If the command `DIR A:` is entered via the keyboard, where does the standard input for the command come from? Where does the standard output go? The input for the command obviously comes from the disk in the A: drive. DIR reads the directory table and displays the output (the list of files) to the console. In a literal sense, the command is

```
DIR A: CON:
```

where A: indicates the standard input and CON: indicates the standard output. This syntax is inaccurate; it is shown only to explain that the output device is "understood" to be the console (specifically, the monitor screen).

Suppose it was important to obtain a complete listing of all files on a disk in printed form. With a short listing, like the example above, the <PrtSc> (Print Screen) Key can be used. Suppose the disk included 70 to 80 different files. The solution is to redirect output from the DIR command to the printer. The symbol for DOS redirection is the greater-than or less-than sign (< or >).

To redirect the output from the DIR command to the printer, we would enter the following command:

```
DIR A:>PRN
```

Remember that PRN is an alias for LPT1:, so it is assumed that a printer is attached to parallel port #1. With the exception of NUL:, it is important to always have a device (printer, modem, etc.) attached to the standard device indicated. If output is redirected to a nonexistent device, the computer will freeze--at best. The results can be even more unpredictable.

Notice that the greater-than and less-than signs used to stipulate redirection serve as pointers, indicating the flow of data. Consider them as the head of an arrow. In the above example, the directory listing (input) is redirected from the A: drive to PRN (the output destination).

If the objective is to create a file in which the directory data are stored, a slightly different version of the DIR command can be used.

```
DIR A:>B:FILELIST.DOC
```

This command would create a new file on the diskette in drive B: that would store all the directory data for files on the disk in the A: drive. In both of the examples, the console screen would not display the directory listing. The output has been redirected away from the monitor screen to another device (PRN) or to a file (B:FILELIST.DOC).

There may be occasions when you need to append data by redirection to an existing file. To understand how to accomplish the appending of new data to an old file, we must first understand how output redirection works. If a file is to be created via redirection, DOS first determines if a file with the same filename exists on the target drive. If it does not, DOS creates the file. However, if the file already exists, DOS sets the write pointer to the beginning of the existing file, and in effect, the old file is overwritten by the redirected data.

For example, if we switched to another disk in drive A: and entered the command

```
DIR A:>B:FILELIST.DOC
```

then the directory data of the files on the new disk would be written to the file FILELIST.DOC, and the previous data would be overwritten.

By using double greater-than or less-than signs (>> or <<), the redirected data are appended to the file rather than overwritten. The important point to remember is that simple redirection to a file (using > or <) will destroy an old file with the same filename. However, the append redirection symbol (using >> or <<) will add the new data to the end of the old file.

GUIDED ACTIVITY

REDIRECT DOS INPUT AND OUTPUT: In this Guided Activity, you will use the DOS redirection capability to redirect first output, then input.

1. The default directory should be C:\DOS or wherever your DOS commands are stored.

2. Type **FORMAT A:/V** and press <ENTER>.

3. Insert a blank unformatted diskette in drive A:, and press <ENTER>.

4. When the format is complete and you are prompted for a volume label, type **UNIT9** and press <ENTER>.

5. When prompted to "Format another (Y/N)?", type **N** and press <ENTER>.

6. Type **CLS** and press <ENTER>.

7. Type **DIR > A:FILELIST** and press <ENTER>.

The previous step generates a directory listing but then redirects it to a file called FILELIST on the disk in the A: drive. Notice that all screen messages were redirected to the file. Your monitor screen did not display the files in the \DOS directory.

8. Type **DIR A:FILELIST** and press <ENTER>.

Figure 9-2 displays the screen response. Notice that a new file resides on your newly formatted data disk. The date and time portion of the directory listing will correspond to the date and time maintained by your computer when the file was created.

```
C:\DOS>DIR > A:FILELIST

C:\DOS>DIR A:FILELIST

 Volume in drive A is UNIT9
 Directory of  A:\

FILELIST         2234   5-06-89  10:20a
       1 File(s)    359424 bytes free

C:\DOS>
```

Fig. 9-2. DIR listing for FILELIST.

9. Type **TYPE A:FILELIST** and press <ENTER>.

Your screen will display the complete directory listing (created by the DIR command) stored earlier in FILELIST.

The end of the TYPE command listing should look like that in Fig. 9-3, except that the number of bytes free may vary--depending on the actual space available on your hard disk.

```
CHKDSK   COM     9850   3-18-87   12:00p
COMP     COM     4214   3-17-87   12:00p
DEBUG    COM    15897   3-17-87   12:00p
DISKCOMP COM     5879   3-17-87   12:00p
DISKCOPY COM     6295   3-17-87   12:00p
EDLIN    COM     7526   3-17-87   12:00p
FIND     EXE     6434   3-17-87   12:00p
GRAFTABL COM     6128   3-17-87   12:00p
GRAPHICS COM     3300   3-17-87   12:00p
JOIN     EXE     8969   3-17-87   12:00p
LABEL    COM     2377   3-17-87   12:00p
MORE     COM      313   3-17-87   12:00p
PRINT    COM     9026   3-17-87   12:00p
RECOVER  COM     4299   3-18-87   12:00p
RESTORE  COM    34643   3-17-87   12:00p
SHARE    EXE     8608   3-17-87   12:00p
SORT     EXE     1977   3-17-87   12:00p
SUBST    EXE     9909   3-17-87   12:00p
TREE     COM     3571   3-17-87   12:00p
BASIC    PIF      369   3-17-87   12:00p
BASICA   PIF      369   3-17-87   12:00p
MORTGAGE BAS     6251   3-17-87   12:00p
       52 File(s)   4900864 bytes free

C:\DOS>
```

Fig. 9-3. File list generated by redirection of FILELIST.

The output redirected to FILELIST can now be redirected to the printer, using the TYPE command and DOS redirection.

10. Be sure your printer is turned on and the paper is set to the top of the form.

11. Type **TYPE A:FILELIST > PRN** and press <ENTER>.

In the following steps, you will use redirection to append another DIR command listing to the file named FILELIST.

12. Type **CD\WEST** and press <ENTER>. This command changes the default drive to the one created in Unit 8 (C:\WEST).

13. Type **CLS** and press <ENTER>.

14. Type **DIR** and press <ENTER>.

Eleven files are displayed, as shown in Fig. 9-4, but three of the listings are really directory entries.

```
C:\WEST>DIR

 Volume in drive C is JONATHAN'S
 Directory of  C:\WEST

.            <DIR>       4-30-89   5:04p
..           <DIR>       4-30-89   5:04p
DEGAS    DOC      4268   5-02-89   7:41p
EAGLE    DOC      1056   5-02-89   7:45p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
TOUCHY   TXT       460   5-02-89   7:18p
TECHNO   TXT      2112   5-02-89   7:00p
TRUCKIN  TXT       713   5-02-89   7:12p
WKSHEETS     <DIR>       5-06-89   7:03a
EMPTY    ONE         42   5-06-89   7:25a
       11 File(s)   4896768 bytes free

C:\WEST>
```

Fig. 9-4. DIR listing of files in the C:\WEST directory.

15. Type **DIR>>A:FILELIST** and press <ENTER>.

The double greater-than sign causes the DIR command output to be appended to the end of FILELIST rather than overwriting the file.

16. Type **TYPE A:FILELIST** and press <ENTER>.

When the screen stops scrolling, it will look like that in Fig. 9-5. The top of the screen shows the end of the original directory listing of directory C:\DOS. The bottom of the screen shows the appended results of the DIR listing for C:\WEST.

```
SORT     EXE     1977   3-17-87  12:00p
SUBST    EXE     9909   3-17-87  12:00p
TREE     COM     3571   3-17-87  12:00p
BASIC    PIF      369   3-17-87  12:00p
BASICA   PIF      369   3-17-87  12:00p
MORTGAGE BAS     6251   3-17-87  12:00p
       52 File(s)   4900864 bytes free

 Volume in drive C is JONATHAN'S
 Directory of  C:\WEST

.            <DIR>      4-30-89   5:04p
..           <DIR>      4-30-89   5:04p
DEGAS    DOC        4268   5-02-89   7:41p
EAGLE    DOC        1056   5-02-89   7:45p
OVERDUE  DOC         617   3-04-89  10:17a
PCAT     DOC         679   4-22-89   7:41p
TOUCHY   TXT         460   5-02-89   7:18p
TECHNO   TXT        2112   5-02-89   7:00p
TRUCKIN  TXT         713   5-02-89   7:12p
WKSHEETS     <DIR>      5-06-89   7:03a
EMPTY    ONE           42   5-06-89   7:25a
       11 File(s)   4894720 bytes free

C:\WEST>
```

Fig. 9-5. Final screen showing results of the TYPE command.

In the remainder of this Guided Activity, you will create a file that, through redirection, will provide input for a DO/S command. Remember that, when using the FORMAT command, three keystrokes and a volume label are required from the keyboard. After invoking the command `FORMAT A:/V`, the user is instructed to insert a disk in the designated drive and to press <ENTER> (first keystroke). After formatting is complete, a volume label is needed, and then the user is asked if another disk should be formatted. The response is either Y for yes or N for no, followed by pressing <ENTER> (final keystroke).

You will use EDLIN to create a text file called INPUTS containing these keystrokes. Then INPUTS will be used through redirection to supply the keystrokes expected by DOS when it executes the FORMAT command.

17. Type **`CD\DOS`** and press <ENTER>.

18. Type **`CLS`** and press <ENTER>.

19. Type **`EDLIN INPUTS`** and press <ENTER>.

The screen should respond with the EDLIN prompt:

```
New file
*
_
```

20. Type **I** and press <ENTER>.

21. Press <ENTER>.

This is the first keystroke expected by the FORMAT command.

22. Type **AUTOFORMAT**, which is the volume label.

23. Press <ENTER>.

24. Type **N**. This responds to the "Format another (Y/N)?" prompt.

25. Press <ENTER>.

26. Press <ENTER>. This is the final keystroke expected by the FORMAT command.

27. Press <CTRL-C>.

28. Type **L**

29. Press <ENTER> to display this short INPUTS file.

The screen should appear as shown in Fig. 9-6.

```
C:\DOS>EDLIN INPUTS
New file
*I
       1:*
       2:*AUTOFORMAT
       3:*N
       4:*
       5:*^C

*L
       1:
       2: AUTOFORMAT
       3: N
       4:
*
```

Fig. 9-6. Screen display while creating the INPUTS file.

30. Type **E**
31. Press <ENTER> to exit and save the INPUTS file.
32. Type **CLS** and press <ENTER>.
33. Leave the disk from the first part of this Guided Activity in drive A:.
34. Type **FORMAT A:/V < INPUTS** (Execution of this command redirects the keystrokes from the INPUTS file to the FORMAT command. Once executed, you lose the ability to intervene and stop the initialization of the disk in drive A:.)

> **NOTE:** Double-check the above line. If you have not typed the command correctly, use the <BACKSPACE> Key to erase incorrect characters, and retype the line before proceeding! Once you press the <ENTER> Key, this command is off and running. By redirecting input from the INPUTS file, you lose control after pressing the <ENTER> Key.

35. Press <ENTER>.

Figure 9-7 shows the screen response. By the time the DOS prompt returns, the disk in drive A: is formatted and assigned a volume label (AUTOFORMAT).

```
C:\DOS>FORMAT A:/V <INPUTS
Insert new diskette for drive A:
and strike ENTER when ready

Format complete

Volume label (11 characters, ENTER for none)? AUTOFORMAT


    362496 bytes total disk space
    362496 bytes available on disk

Format another (Y/N)?N
C:\DOS>
```

Fig. 9-7. Screen display of the FORMAT command using redirection from the INPUTS file.

36. Type **CLS** and press <ENTER>.

37. Type **DIR A:** and press <ENTER>. Notice the volume label is AUTOFORMAT.

PIPES AND FILTERS--DOS BUILDING BLOCKS

In home construction, pipes are used to convey material (gas, air, water) from one location to another. Filters, on the other hand, are used to alter material passing through the home. A furnace filter eliminates dust from the forced-air system; thus, it alters the material (air) being piped throughout the house.

DOS pipes and filters function in parallel ways. Piping allows the output from one program or command to be used as the input for another program or command. In a sense, it directs the standard output to become the input for another operation. The programs or commands become chained together. A pipe is indicated using the | character.

> **NOTE:** Please note that the piping character usually appears on the keyboard and screen as a vertical broken line--rather than the solid vertical line used in this text.

For clarification, there is a difference between piping and redirection. Piping can serve as a connection between DOS commands, i.e., output from the first command becomes the input for the second command. It can also connect programs or subroutines. Redirection deals only with devices. The output of data may be redirected to the printer, a modem, or the console. You cannot redirect data to another DOS command.

#9

The DOS piping operation creates one or more temporary files in the root directory of the default drive. DOS 2.0 names these files %PIPEx.$$$, where x is an integer. DOS versions 3.0 and higher create files using an eight-digit hexadecimal number based on output from the IBM PC's internal clock. It is important that these files not be erased until the piping operation is complete.

DOS filters accept data from standard input sources and then modify the data. The three standard DOS filters are SORT, MORE, and FIND.

THE SORT FILTER

The most commonly used and easily understood filter is the SORT filter. SORT is an external DOS command that manipulates text data that it receives as input and then outputs the data in sorted form. With DOS 2.x and earlier versions, the sort is performed using the ASCII collating sequence. Beginning with DOS 3.x, a modified ASCII collating sequence is used.

In the ASCII collating sequence, each alphabetical and numerical character is assigned a binary value. There are significant differences between the ASCII text sort and a regular alphabetic or numeric sort.

The left column below lists selected characters in random order, while the right column lists the same characters after an ASCII text sort such as would be generated by DOS 2.x or earlier.

Random Elements	DOS 2.x Sort
junk5mail	#
20	%
z	1
Apple	10
710	101
APPLE	20
Zoo	710
10	99
junkmail	APPLE
1	Apple
junk mail	Zoo
101	apple
99	junk mail
%	junk5mail
apple	junkmail
#	z

By examining the above example, it should become obvious that the ASCII text sort more closely approximates an alphabetic rather than numeric sort. However, the results are still puzzling until you recognize that uppercase alphabetic characters are considered completely different characters from lowercase characters. Thus the lowercase letter "a" follows the uppercase letter "Z" in this type of sort. Characters are evaluated beginning with the leftmost character unless directed otherwise. Numbers are not necessarily put in numeric order. Remember, each character place is assigned an ASCII value and then sorted based on that value. The range of values is outlined in the following table:

Character(s)
(in ASCII collating sequence)

#11

space (considered as a character)
special characters (@, #, $, %, ^, &, etc.)
numbers (in ascending order: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
uppercase letters (A, B, C, D, E, F, G, H, etc.)
lowercase letters (a, b, c, d, e, f, g, h, etc.)

Each special character and each non-printing character (<SHIFT> Key, <CTRL> Key, <ALT> Key) are assigned an ASCII value. If the ASCII text sort is not understood, then the DOS 2.x SORT filter will generate some unexpected results.

Effective with DOS 3.1, a modification was made to the sort based on binary values. The ASCII collating sequence is still used, with one exception. Lowercase letters a-

z are given equal value to uppercase letters A-Z. The following table demonstrates the effect of the different sorts:

Random Elements	DOS 2.x SORT	DOS 3.x SORT
APPLE	APPL	APPL
APPL	APPLE	APPLE
apple	LIME	apple
orange	apple	LIME
LIME	orange	orange

For the duration of our discussion of the SORT filter, we'll assume use of DOS 3.1 or higher.

In a previous example, we wrote directory information from a disk directly to the printer. That command form was

```
DIR A:>PRN
```

For the next few examples, we'll use the following directory listing:

FORMLTR	DOC	14496	10-22-86	8:12p
HMBUDGET	WS	11717	9-14-85	12:47a
ADDRESS	DBF	98373	8-15-86	10:30p
ADDRESS	NDX	963	8-15-86	10:30p
ADDRESS	DBS	1123	8-15-86	10:30p
FORMLTR	BDC	13888	10-09-86	7:22p
HMBUDGET	BWS	9124	8-13-85	1:05p

The command used to generate a random listing of files could be modified to print an ASCII text sorted directory. Assuming that the A: drive is the default drive, the command would be entered as

```
DIR|SORT>PRN
```

This command takes the output from the DIR command and "pipes" it to the SORT filter. The filter modifies the directory listing into an alphabetical sort. The results are then redirected (>) to the printer (PRN). If the default drive were not A:, then the command would be entered as

```
DIR A:|SORT>PRN
```

Either command would generate a printed report like the following:

```
ADDRESS     DBF     98373     8-15-86     10:30p
ADDRESS     DBS      1123     8-15-86     10:30p
ADDRESS     NDX       963     8-15-86     10:30p
FORMLTR     BDC     13888     10-09-86     7:22p
FORMLTR     DOC     14496     10-22-86     8:12p
HMBUDGET    BWS      9124     8-13-85      1:05p
HMBUDGET    WS      11717     9-14-85     12:47a
```

THE SORT PARAMETERS

The SORT filter includes two optional parameters. The sort can be created in reverse ASCII text sort order; that is, 9 is sorted before 1 and Z before A. The syntax of the command would be

```
SORT/R
```

The second SORT parameter stipulates the column on which the sort will begin. Without this parameter, DOS assumes the sort to begin in column 1. The syntax of the second parameter is

```
SORT/+n
```

where n represents an integer.

The directory listing generated by the DIR command lists bits of data beginning in specific columns. Beginning column numbers for each bit of data are listed in the following table:

Data	Column Number
Filename	1
Extension	10
File size	14
Date created	24
Time created	34

By knowing the beginning column locations, the output of the DIR command may be sorted by filename, extension, file size, date created, or time created. Though it may not be very helpful, a directory of files by creation date in reverse ASCII text sort order may be generated by using the command form

```
DIR A:|SORT/R/+24>PRN
```

A slightly more useful form would be to sort by file size using

```
DIR A:|SORT/+14>PRN
```

It should be noted that if you sort by column 24 (Date created) or 34 (Time created), the sort is not in chronological order. Rather, the sort treats the data as a text string and sorts according to the ASCII collating sequence. For example, the following dates are in ASCII order but not chronological order:

01-01-89
01-02-88
01-05-88
01-06-88
03-24-87

Likewise, the following times are in ASCII collating sequence order, but not in chronological order. Remember that the ASCII sort starts with the leftmost character and works to the right while ordering them. Therefore in the above list, the 01 (first two columns) is evaluated before the 87 or 88 (last two columns) is ever reached. In the following list, the hour is evaluated and sorted before the a.m. (a) or p.m. (p) is ever reached:

1:00a
1:29a
2:16p
2:19a
3:18p
6:39a

THE MORE FILTER

In the earlier discussion of the DIR command, the /P parameter was explained. It generates a screenful of directory listings at a time and then pauses until you strike another key. The MORE filter behaves in a similar manner. It breaks data from the standard input device into blocks of one screen page each. After displaying a full page of output, the message "--More--" appears on the screen. Pressing any key causes the next screenful of data to be displayed. The process will continue until all subsequent pages of data are displayed.

The TYPE command is often used to glance at the contents of an ASCII text file. A common problem in using the TYPE command is that the text scrolls off the screen before it can be read. A practical solution is to use the TYPE command in combination with the MORE filter. The following command form would display one screenful of the text file, LETTER.TXT, at a time:

```
TYPE A:LETTER.TXT|MORE
```

Another situation where the MORE filter can be handy is in the use of the CHKDSK command with the /V parameter. Remember that the CHKDSK/V command checks each file and displays the filename and its path. When this command is used on a hard disk or a floppy disk with numerous files, the data scrolls off the screen too fast to read. You can combine the MORE filter with the

command to view one screenful of data at a time. To execute the command on drive C:, the command form would be

```
CHKDSK C:/V|MORE
```

GUIDED ACTIVITY

USE THE SORT AND MORE FILTERS: In the last Guided Activity, you created a file called FILELIST and then reformatted that disk with the AUTOFORMAT label. In this Guided Activity, you will first recreate that FILELIST file and then proceed to sort the data in that file. Since the DOS files are typically installed on your hard disk in alphabetical order, this exercise will use the SORT filter coupled with the FILELIST file to list DOS external commands in reverse alphabetical order.

1. Be sure the disk with the AUTOFORMAT label is still in drive A:.

2. Type **CLS** and press <ENTER>.

3. The default subdirectory should still be C:\DOS; if not, change to that subdirectory.

4. Type **DIR>A:FILELIST** and press <ENTER>.

5. Type **DIR A:** and press <ENTER>. The volume label should be AUTOFORMAT, and the disk should contain a single file (FILELIST).

You may notice that FILELIST is 2275 bytes, a little larger than what was shown in Fig. 9-2. The new FILELIST file contains an additional listing (for the INPUTS file) in the C:\DOS directory. Listing the details of that file consumes 41 additional bytes in the FILELIST file.

6. Type **TYPE A:FILELIST** and press <ENTER>. Notice that the listing for the INPUTS file is at the end of the FILELIST text.

7. Type **A:** and press <ENTER> to switch to drive A:.

8. Type **CLS** and press <ENTER>.

9. Type **TYPE FILELIST|SORT/R** and press <ENTER>.

The screen will scroll the DOS filenames past in reverse order. When the scrolling ends, the bottom of your screen will list the files shown in Fig. 9-8.

```
DEBUG    COM     15897   3-17-87   12:00p
COUNTRY  SYS     11285   3-17-87   12:00p
COMP     COM      4214   3-17-87   12:00p
COMMAND  COM     25307   3-17-87   12:00p
CHKDSK   COM      9850   3-18-87   12:00p
BASICA   PIF       369   3-17-87   12:00p
BASICA   COM     36403   3-17-87   12:00p
BASIC    PIF       369   3-17-87   12:00p
BASIC    COM      1063   3-17-87   12:00p
BACKUP   COM     31913   3-18-87   12:00p
ATTRIB   EXE      9529   3-17-87   12:00p
ASSIGN   COM      1561   3-17-87   12:00p
APPEND   EXE      5825   3-17-87   12:00p
ANSI     SYS      1678   3-17-87   12:00p
5202     CPI       459   3-17-87   12:00p
4201     CPI     17089   3-18-87   12:00p
..             <DIR>     5-25-87    3:34p
.              <DIR>     5-25-87    3:34p
 Volume in drive C is JONATHAN'S
 Directory of  C:\DOS
       53 File(s)   3919872 bytes free


A:\>
```

Fig. 9-8. End of screen display using the reverse sort on the FILELIST file.

We can also use the DIR command with the /R parameter.

10. Type **C:** and press <ENTER>.

11. Type **CD\DOS** and press <ENTER> to move to the directory with the DOS commands.

12. Type **DIR C:\DOS|SORT/R** and press <ENTER>.

The reverse DIR listing generated by the last step scrolls through a long list of filenames. When the scrolling list stops, the screen will look like that in Fig. 9-9. This listing is identical to the listing in Fig. 9-8, except for the bytes free. That is because the bytes used in Fig. 9-8 include storage for the two temporary files created by the SORT filter. Even though SORT used temporary files to create the listing in Fig. 9-9, the files were erased before the data were displayed on the screen.

```
DEBUG    COM     15897   3-17-87   12:00p
COUNTRY  SYS     11285   3-17-87   12:00p
COMP     COM      4214   3-17-87   12:00p
COMMAND  COM     25307   3-17-87   12:00p
CHKDSK   COM      9850   3-18-87   12:00p
BASICA   PIF       369   3-17-87   12:00p
BASICA   COM     36403   3-17-87   12:00p
BASIC    PIF       369   3-17-87   12:00p
BASIC    COM      1063   3-17-87   12:00p
BACKUP   COM     31913   3-18-87   12:00p
ATTRIB   EXE      9529   3-17-87   12:00p
ASSIGN   COM      1561   3-17-87   12:00p
APPEND   EXE      5825   3-17-87   12:00p
ANSI     SYS      1678   3-17-87   12:00p
5202     CPI       459   3-17-87   12:00p
4201     CPI     17089   3-18-87   12:00p
..             <DIR>     5-25-87    3:34p
.              <DIR>     5-25-87    3:34p
 Volume in drive C is JONATHAN'S
 Directory of  C:\DOS
       53 File(s)   3915776 bytes free



C:\DOS>
```

Fig. 9-9. Reverse DIR listing of the \DOS subdirectory.

The results in Fig. 9-9 may be generated with an abbreviated form of the command--as shown in the next step.

13. Type **DIR|SORT/R** and press <ENTER>.

#13

If you were to interrupt the sort process midstream, you would see two new files. They are temporary files created by the SORT command. If you type DIR for a directory listing, the two temporary files will not be displayed. They are erased automatically after SORT is through using them.

In each of the preceding examples, the directory listing scrolls off the screen. The MORE filter can be used to generate one page of screen display at a time. The user must then press any key to proceed to the next page of screen display.

14. Type **DIR|SORT/R|MORE** and press <ENTER>.

15. Each time the "--More--" prompt appears at the bottom of the screen, press any key--until the DOS prompt returns.

To find the temporary files created by SORT and MORE, you must check the root directory. Of course, the DIR command will not show them. By using the MORE command you can stop the sort in midstream before the files are erased.

16. Type **CD** and press <ENTER> to change to the root directory.

17. Type **DIR|SORT/R|MORE** and press <ENTER>.

18. Press any key to see all the screenfuls of text. If necessary, continue pressing a key until the DOS prompt appears. The final screen will list a group of files similar to that shown in Fig. 9-10.

```
JOY        BAS         115   12-26-86    9:30a
INSET      BAT          23    8-31-87   10:54a
INSET          <DIR>          1-01-80   12:19a
-- More --

HS         BAT          17    8-22-87    7:05a
HIJAAK         <DIR>          4-25-89    9:40p
GAMES      BAT          11    7-08-87    6:20a
FRECOVER DAT         38912    3-22-89   11:23a
FRECOVER BAK         38912    3-22-89   11:23a
FONTS      BAT          29    9-05-87   10:33a
DOS            <DIR>          5-25-87    3:34p
CONFIG     SYS          78    4-20-89    6:29p
COMMAND    COM       25307    3-17-87   12:00p
BACKUP     LOG          61   10-26-88    3:51p
AUTOEXEC BAT           107    4-20-89    6:30p
0A3B1C18               0    5-06-89   10:59a
0A3B1C13               0    5-06-89   10:59a
 Volume in drive C is JONATHAN'S
 Directory of  C:\
       37 File(s)   4954112 bytes free


C:\>
```

Fig. 9-10. Final screenful of text generated by the MORE filter.

The last two files on the monitor screen are temporary ones created by the SORT and MORE commands. The files used by the SORT command are named 0A3B1C18 and 0A3B1C13. They are given names by DOS based on the computer's internal clock. When DOS captured the directory listing, the byte size of the temporary files was zero. The date and time used to name the temporary files depends on when you actually executed the SORT command. These temporary files are always stored in the root directory.

Next, you will sort using the column parameter. Column 14 is the beginning of the file size data (in bytes) from the directory. To work with a smaller list of files, you will change to the \WEST directory.

19. Type **CD\WEST** and press <ENTER>.

20. Type **CLS** and press <ENTER>.

21. Type **DIR** and press <ENTER>.

Figure 9-11 displays the file listing for the \WEST directory.

```
C:\WEST>DIR

 Volume in drive C is JONATHAN'S
 Directory of  C:\WEST

.            <DIR>       4-30-89   5:04p
..           <DIR>       4-30-89   5:04p
DEGAS    DOC     4268    5-02-89   7:41p
EAGLE    DOC     1056    5-02-89   7:45p
OVERDUE  DOC      617    3-04-89  10:17a
PCAT     DOC      679    4-22-89   7:41p
TOUCHY   TXT      460    5-02-89   7:18p
TECHNO   TXT     2112    5-02-89   7:00p
TRUCKIN  TXT      713    5-02-89   7:12p
WKSHEETS     <DIR>       5-28-89   4:46p
EMPTY    ONE        42   5-12-89   5:46a
       11 File(s)   3919872 bytes free

C:\WEST>
```

Fig. 9-11. DIR listing for C:\WEST.

These files are in random order. In the following steps, you will reorder or sort these files by byte size and extension.

22. Type **CLS** and press <ENTER>.

23. Type **DIR|SORT/+14** and press <ENTER>.

#13

```
C:\WEST>DIR|SORT/+14


EMPTY    ONE        42   5-06-89   7:25a
TOUCHY   TXT       460   5-02-89   7:18p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
TRUCKIN  TXT       713   5-02-89   7:12p
EAGLE    DOC      1056   5-02-89   7:45p
TECHNO   TXT      2112   5-02-89   7:00p
DEGAS    DOC      4268   5-02-89   7:41p
 Directory of  C:\WEST
.            <DIR>       4-30-89   5:04p
..           <DIR>       4-30-89   5:04p
WKSHEETS     <DIR>       5-06-89   7:03a
       11 File(s)   4952064 bytes free
 Volume in drive C is JONATHAN'S

C:\WEST>
```

Fig. 9-12. DIR listing sorted by data beginning in column 14.

Remember that the sort is not strictly by file size, but rather by the data in column

24. In most cases that column contains the file size, although not always--as is the case for the last six lines.

25. Type **CLS** and press <ENTER>.

26. Type **DIR|SORT/+10** and press <ENTER>. Filename extensions begin in column 10.

#13

```
C:\WEST>DIR¦SORT/+10


.           <DIR>          4-30-89   5:04p
..          <DIR>          4-30-89   5:04p
WKSHEETS    <DIR>          5-06-89   7:03a
      11 File(s)   4950016 bytes free
OVERDUE  DOC        617   3-04-89  10:17a
PCAT     DOC        679   4-22-89   7:41p
EAGLE    DOC       1056   5-02-89   7:45p
DEGAS    DOC       4268   5-02-89   7:41p
 Volume in drive C is JONATHAN'S
EMPTY    ONE         42   5-06-89   7:25a
TOUCHY   TXT        460   5-02-89   7:18p
TRUCKIN  TXT        713   5-02-89   7:12p
TECHNO   TXT       2112   5-02-89   7:00p
 Directory of  C:\WEST

C:\WEST>
```

Fig. 9-13. DIR listing sorted by date beginning in column 10.

This sort may look unusual until you remember that the SORT command works on data starting in column 10. It is not technically sorting by filename extension. Notice that column 10 includes these characters:

space	in the 10th column of the first four lines
D	from the DOC extension
n	from the phrase "Volume in drive"
O	from the ONE extension
T	from the TXT extension
y	from the word "Directory"

Each line is in the proper order based on the character in column 10.

27. Type **CLS** and press <ENTER>.

THE FIND FILTER

The third and final DOS filter is the FIND command. The FIND filter seeks out occurrences of a specified text string in specified file(s). This external DOS command follows the syntax

```
FIND "textstring" [d:][path][filename.ext]
```

If we sought to locate all occurrences of the phrase "Strategic Defense Initiative" in a file called EINSTEIN.DOC on the A: drive, then the command would be executed as

```
FIND "Strategic Defense Initiative" A:EINSTEIN.DOC
```

Without adding one of three possible parameters, the results would duplicate each line in which the text string appeared. The results are shown in Fig. 9-14.

```
C:\>FIND "Strategic Defense Initiative" A:EINSTEIN.DOC

---------- A:EINSTEIN.DOC
writing to you regarding a program called "Strategic Defense Initiative".
Yes, "Strategic Defense Initiative" is the same as what has affectionately
cannot support "Strategic Defense Initiative" as it is currently defined.

C:\>
```

Fig. 9-14. Results of the FIND filter.

It is important to note that the FIND filter is case-sensitive. The text string used as a parameter for the filter must be identical to the text sought. In other words, "Strategic Defense Initiative" will not match with "strategic defense initiative" or

even "Strategic defense Initiative." The text string in the command form must be enclosed in quotation marks.

Examine the following three text phrases. Assume this text is part of a document named A:EINSTEIN.DOC.

```
1:    I was initially opposed to the Strategic Defense Initiative.
2:    However, Warthog's fine expose of SDI in the article
3:    "Strategic Defense Initiative" changed my mind.
```

If you use the command form

```
FIND "Strategic Defense Initiative" A:EINSTEIN.DOC
```

you will locate both occurrences of Strategic Defense Initiative (lines 1 and 3). You cannot use the FIND filter to locate only a text string enclosed in quotation marks within the document. The following command form is incorrect and would return the error message "Parameter format not correct":

```
FIND ""Strategic Defense Initiative""
     A:EINSTEIN.DOC
```

Multiple files may be searched using a single FIND command. If we sought all occurrences of "Strategic Defense Initiative" in files named EINSTEIN.DOC, STARWARS.TXT, and SDI.DOC, the command would take the form

```
FIND "Strategic Defense Initiative" A:EINSTEIN.DOC
     A:SDI.DOC A:STARWARS.TXT
```

The resulting screen display might appear like that shown in Fig. 9-15, assuming that there were no occurrences of the text string in the SDI.DOC file.

```
C:\>FIND "Strategic Defense Initiative" A:EINSTEIN.DOC A:SDI.DOC A:STARWARS.TXT

---------- A:EINSTEIN.DOC
writing to you regarding a program called "Strategic Defense Initiative".
Yes, "Strategic Defense Initiative" is the same as what has affectionately
cannot support "Strategic Defense Initiative" as it is currently defined.

---------- A:SDI.DOC

---------- A:STARWARS.TXT
The Technical title is "Strategic Defense Initiative."
More often the "Strategic Defense Initiative" is referred to as "STARWARS" by

C:\>
```

Fig. 9-15. Results of the FIND filter search through multiple files.

Unfortunately, global filename characters such as the asterisk and question mark cannot be used in filenames or extensions that are used with the FIND filter. This is a severe limitation, particularly when you are searching through numerous files that might include the sought-after text string.

THE FIND FILTER PARAMETERS

The FIND command lists only the line of text on which the sought-after text string appears. If no occurrences of the text string have been found, the display will list the filename but nothing else. There may be times when knowing the relative line number would be helpful. The /N parameter lists the relative line number at the beginning of the displayed line in which the sought-after text occurs. When determining the relative line number, DOS begins counting at the first line in the file. Blank lines are counted also. Figure 9-16 is an example of the listing created using the /N parameter.

```
C:\>FIND/N "Strategic Defense Initiative" A:EINSTEIN.DOC

---------- A:EINSTEIN.DOC
[17]writing to you regarding a program called "Strategic Defense Initiative".
[24]Yes, "Strategic Defense Initiative" is the same as what has affectionately
[35]cannot support "Strategic Defense Initiative" as it is currently defined.

C:\>
```

Fig. 9-16. Results of a search using FIND with the /N parameter.

Two additional parameters or switches may be used with the FIND command. The basic FIND command syntax searches through the named file(s) for lines of text that include the text string. This process can be reversed by using the /V parameter. The command

```
FIND/V "the" LETTER.TXT LETTER3.DOC
```

will list to the monitor screen all lines in the files LETTER.TXT and LETTER3.DOC that do not include the word "the".

The final parameter is the /C or count parameter. Rather than displaying the lines containing the text string, the /C parameter displays the number of lines that include the sought-after string. If we modified the command used to find occurrences of "Strategic Defense Initiative" to generate only a line count, the command would be

```
FIND/C "Strategic Defense Initiative"
     A:EINSTEIN.DOC A:SDI.DOC  A:STARWARS.TXT
```

The screen display would list the messages shown in Fig. 9-17.

```
C:\>FIND/C "Strategic Defense Initiative" A:EINSTEIN.DOC A:SDI.DOC A:STARWARS.TX
T

---------- A:EINSTEIN.DOC: 3

---------- A:SDI.DOC: 0

---------- A:STARWARS.TXT: 2

C:\>
```

Fig. 9-17. Results of a search using FIND with the /C parameter.

Even though the last T in .TXT wrapped to a second line, DOS still executes the command correctly.

#15 While the /V and /N parameters may be combined, the /C parameter cannot be combined with any other switch on a single execution of the FIND command. If /C is used, the other switches are ignored.

GUIDED ACTIVITY

USE THE FIND FILTER: In this Guided Activity, you will create a text file using EDLIN and then use the FIND filter to locate occurrences of specific words within the file.

1. Be sure that your data disk with the AUTOFORMAT volume label (from the last exercise) is in drive A: and your default drive and path is C:\DOS.

2. Type **EDLIN A:FINDTEXT**

3. Press <ENTER>.

The screen will respond with the EDLIN prompt:

```
New file
*
_
```

4. Type `I`
5. Press <ENTER>.
6. Type `This is a file in which to`
7. Press <ENTER>.
8. Type `test the FIND Filter provided`
9. Press <ENTER>.
10. Type `by DOS 3.0 and higher.`
11. Press <ENTER>.
12. Press <ENTER> again.
13. Type `To test or not to test...`
14. Press <ENTER>.
15. Type `that is not the question.`
16. Press <ENTER>.
17. Type `Rather, the question is whether`
18. Press <ENTER>.
19. Type `Find can find the word test`
20. Press <ENTER>.
21. Type `or if the test will be failed.`
22. Press <ENTER>.
23. Press <ENTER> again.
24. Type `TEST PATTERN!`
25. Press <ENTER>.

26. Press <CTRL-C>.

Your screen should appear like the one shown in Fig. 9-18.

```
C:\DOS>EDLIN A:FINDTEXT
New file
*I
       1:*This is a file in which to
       2:*test the FIND Filter provided
       3:*by DOS 3.0 and higher.
       4:*
       5:*To test or not to test...
       6:*that is not the question.
       7:*Rather, the question is whether
       8:*Find can find the word test
       9:*or if the test will be failed.
      10:*
      11:*TEST PATTERN!
      12:*^C

*
```

Fig. 9-18. EDLIN test file text.

If the text in your file is not identical to that of Fig. 9-18, use EDLIN to go back and correct.

27. Type **E** to exit, and press <ENTER> to save the file FINDTEXT.

28. Type **CLS** and press <ENTER>.

29. Type **FIND "test" A:FINDTEXT** and press <ENTER>.

The search conducted by the FIND filter will display the messages shown in Fig. 9-19.

```
C:\DOS>FIND "test" A:FINDTEXT

---------- A:FINDTEXT
test the FIND Filter provided
To test or not to test...
Find can find the word test
or if the test will be failed.

C:\DOS>
```

Fig. 9-19. Results of the FIND search for "test".

30. Type **FIND "Test" A:FINDTEXT** and press <ENTER>.

The FIND filter is case-sensitive, so no occurrences of "Test" are found in the file. The screen responds with

---------- A:FINDTEXT

meaning that the specified text was not found.

31. Type **FIND "TEST" A:FINDTEXT** and press <ENTER>.

The screen response to this search appears in Fig. 9-20.

```
C:\DOS>FIND "test" A:FINDTEXT

---------- A:FINDTEXT
test the FIND Filter provided
To test or not to test...
Find can find the word test
or if the test will be failed.

C:\DOS>FIND "Test" A:FINDTEXT

---------- A:FINDTEXT

C:\DOS>FIND "TEST" A:FINDTEXT

---------- A:FINDTEXT
TEST PATTERN!

C:\DOS>
```

Fig. 9-20. Results of three FIND searches for "test", "Test", and "TEST".

32. Type **CLS** and press <ENTER>.

33. Type **FIND/V "test" A:FINDTEXT** and press <ENTER>.

This command form displays the lines that do not contain the "test" text string.

The results of the search for lines without the text "test" are shown in Fig. 9-21.

```
C:\DOS>FIND/V "test" A:FINDTEXT

---------- A:FINDTEXT
This is a file in which to
by DOS 3.0 and higher.

that is not the question.
Rather, the question is whether

TEST PATTERN!

C:\DOS>
```

Fig. 9-21. Results of the FIND search for lines without "test".

34. Type **FIND/C "test" A:FINDTEXT** and press <ENTER>.

The monitor screen will display the results of the search as

---------- A:FINDTEXT: 4

There are four lines in the file FINDTEXT that include at least one occurrence of the specified text "test".

The final step will indicate the relative line numbers where the text "test" is found in the file.

35. Type **CLS** and press <ENTER>.

36. Type **FIND/N "test" A:FINDTEXT** and press <ENTER>.

The results of this command form are displayed in Fig. 9-22.

```
C:\DOS>FIND/N "test" A:FINDTEXT

---------- A:FINDTEXT
[2]test the FIND Filter provided
[5]To test or not to test...
[8]Find can find the word test
[9]or if the test will be failed.

C:\DOS>
```

Fig. 9-22. Results of the FIND search using the /N parameter.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`DIR A: >B:LIST.TXT`	Creates an ASCII text file on drive B: called LIST.TXT by redirection of the output from the DIR command.
`DIR >PRN`	Redirects the output from the DIR command to a printer connected to parallel port one.
`DIR|MORE`	Generates a listing of all files on the default drive, one screenful at a time.
`DIR|SORT`	Generates an alphabetical listing of all files on the default drive.
`DIR|SORT/R`	Generates a listing of all files on the default drive in reverse alphabetical order.
`DIR|SORT>PRN`	Generates a listing of all files on the default drive in alphabetical order and redirects the directory listing to the printer attached to LPT1: instead of the monitor screen.
`FIND "Germany" A:COUNTRY.DOC`	Locates all occurrences of the word "Germany" in the file COUNTRY.DOC on the disk in drive A:.
`FIND/N "Germany" A:COUNTRY.DOC`	Locates all occurrences of the word "Germany" in the file COUNTRY.DOC on the disk in drive A: and indicates the relative line number on which each occurrence appears.
`TYPE FILEONE.TXT >PRN`	Redirects the listing created by the TYPE command to a printer connected to parallel port one.

REVIEW QUESTIONS

1. Explain the difference between a serial and a parallel port. Why is this a concern when discussing DOS standard devices?
2. How many DOS standard devices are there? List them.
3. Which standard device(s) is/are considered the default device(s)?
4. Which standard device is typically used for debugging? How would a programmer use this device?
5. If standard devices did not exist, other than the default device, how would computing be limited?
6. What does redirection of standard input and output mean? Give an example.
7. When redirecting output to a disk file, how can the output be appended to an existing file?
8. Define DOS pipes and filters. What is the difference?
9. What is the difference between a pipe and redirection?
10. What is the ASCII collating sequence?
11. What is the difference between the DOS 2.x and 3.x SORT filters?
12. In what order would the following filenames be sorted by DOS 2.x? By DOS 3.x?

    Zebra
    cat
    Antelope
    939
    ANTELOPE
    CATwalk
    129
    93
13. Which parameters are used with the SORT filter?
14. Which parameters are used with the MORE filter?
15. Which parameters are used with the FIND filter?

UNIT EXERCISES

1. Create a batch file called ALPHA.BAT that displays a directory listing of all files in alphabetical order by the filename.

2. Create a batch file called BYDATE.BAT that displays a directory listing of all files in order of creation date.

DOCUMENTATION RESEARCH

The answers to this unit's questions may be found in Chapter 7 of the *IBM Disk Operating System Version 3.30 Reference* manual or in Chapter 3 of the *Microsoft MS-DOS User's Reference*.

1. Are there parameters that can be used with the MORE command?

2. What is the maximum file size that can be ordered with the SORT filter?*

3. If the /+n parameter is not used with the SORT filter, what default value does n assume?

4. Can the input and output filenames used with the SORT filter be identical?*

5. Can redirection (>) be used to send output to the MORE filter?

* Not covered in the MS-DOS documentation.

APPLICATION G

PART 1: SEARCHING FOR TEXT PHRASES IN A FILE

In Unit 9, we discussed redirection and DOS filters. In this application, you will search a text file for all occurrences of a specific phrase and redirect the results to a printer.

A. Obtain a copy of the file SEARCH.TXT from your instructor. This file is an ASCII text file.

B. Use a DOS filter to locate all occurrences of the phrase *He should have known better.* The period should be included in the search.

C. Redirect the results of the search to a printer.

D. Repeat the above search for the phrase *He should have known better.* (including the period). However, this time you should redirect (to a printer) a listing of relative line numbers where the phrase occurs in the file.

E. Turn in a printed copy of the results to your instructor.

PART 2: USING PIPES, FILTERS, AND REDIRECTION

In this application, you will use the pipes, filters, and redirection learned in Unit 9. All files for this application are on the West Student Data Disk.

A. Find all occurrences of the word *Geostar* in the file TRUCKIN.TXT. Redirect the results displayed on the screen to your printer. The printed report should be handed in to your instructor.

B. Find all occurrences of the following words in the file TECHNO.TXT.

self-service
first
computer

Redirect the results displayed on the screen to your printer. The printed report should be handed in to your instructor.

C. Display the TECHNO.TXT file on your monitor screen one page at a time. Demonstrate to your instructor that you can accomplish this task.

UNIT 10

SCREEN AND KEYBOARD CONTROL COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. list the five types of uses of the MODE command;
2. list the MODE command forms used to control the screen and keyboard;
3. discuss code page switching and country codes;
4. explain the uses of the ANSI.SYS device driver;
5. discuss the use of escape sequences;
6. describe how the PROMPT command accesses ANSI.SYS functions.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. ANSI.SYS
2. CHCP
3. COUNTRY.SYS
4. MODE
5. NLSFUNC
6. PROMPT
7. SELECT.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): code page and CONFIG.SYS file.
____Guided Activity: Use the PROMPT Command and ANSI.SYS to Move the Cursor.
____Guided Activity: Use the PROMPT Command to Alter Screen Colors and Attributes.
____Guided Activity: Redefine Keyboard Keys with ANSI.SYS and the PROMPT Command.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

FIVE MODES OF OPERATION

The MODE command is used to accomplish five seemingly unrelated functions, though each does affect the mode of operation of the microcomputer system. The MODE command has five options or formats that enable you to

1. switch the monitor display mode;
2. prepare a device for code page switching;
3. set protocol for a serial port;
4. set the format of printer output;
5. redirect parallel port output to a serial port or cancel the redirection.

The first two functions of the MODE command will be covered in this unit. The other functions will be covered in Units 11 and 12.

NOTE: The results and actions of the MODE command vary greatly with the hardware in use. In fact, the primary use of the command is to adapt the operating system to the equipment on hand. For that reason, it is difficult to demonstrate or guide the user through specific projects with most of the MODE command functions. Out of necessity, sections on the MODE command concentrate more on an explanation of the function and less on step-by-step exercises.

THE MODE COMMAND--CHANGING DISPLAY ADAPTERS

The first MODE command option is to set the kind of display monitor adapter expected by the system and indicate the number of characters displayed vertically across the screen. The syntax of this option is

```
MODE [mode][,s][,t]
```

where mode represents the display adapter type and characters across the screen (see the following list). The s would be replaced with either an R or an L to shift the screen image, respectively, to the right or left. The t parameter generates a test pattern.

The display adapter/width options are listed below.

BW40	Disables color after switching to the color/graphics display adapter and generates 40 characters per line across the screen in monochrome.
BW80	Disables color after switching to the color/graphics display adapter and generates 80 characters per line across the screen in monochrome.
40	Displays 40 characters per line on the color/graphics display adapter.
80	Displays 80 characters per line on the color/graphics display adapter.
CO40	Enables color after switching to the color/graphics display adapter and generates 40 characters per line across the screen.
CO80	Enables color after switching to the color/graphics display adapter and generates 80 characters per line across the screen.
MONO	Causes the monochrome display adapter to be the active display adapter and generates 80 characters per line across the screen. There is no 40-character option with the monochrome display adapter.

If you omit the last two parameters, you can simply switch display modes. For instance, you could switch to 40-column color display by entering at the system prompt

```
MODE CO40
```

To switch back to 80-column color display, simply enter

```
MODE CO80
```

The following option of the MODE command does shift the screen display slightly to the right or left. To switch right, the R parameter is used, while L switches the display to the left. The type of display adapter must be stipulated in the command.

If you are working with a monochrome display adapter and monitor, the form of these commands would be

```
MODE MONO,R
MODE MONO,L
```

In the 40-column mode, the R or L shifts the screen one character in the specified direction. In 80-column mode, the shift is two characters to the right or left, as indicated. If you cannot shift to the right or left, an error message will be displayed on the screen. For instance, if the screen text could not be adjusted to the right, you would see the message

Unable to shift screen right

If you try to use the MONO mode on a color display (or vice versa), DOS will issue the error message

Function not supported on this computer - mode

The "mode" is replaced in the error message with the non-supported mode (such as MONO). To generate a test pattern for a color/graphics display and shift the screen right, the following command form may be used:

```
MODE CO80,R,T
```

When combining the T parameter with a right or left shift, a screen message will ask if the monitor image is properly aligned. If you respond with no, the screen will shift again in the direction originally indicated. A yes response concludes the command, and the test pattern disappears. The test pattern is not supported by all hardware. If you do not get a test pattern, your equipment may not be fully supported by the DOS 3.x MODE command.

CODE PAGE SWITCHING

The word "International" in International Business Machines (IBM) Corporation indicates that DOS is not just a North American product. MS-DOS and PC DOS provide national language support through the use of **code pages**. If you are familiar with any foreign language, such as Spanish, French, Portuguese, German, or the like, you are probably aware that those languages use characters and accent marks not found in English. Through the use of various code pages, DOS enables your system hardware to display and print non-English language characters.

The computer assigns a numeric value to each keyboard character. When data is stored, those numeric values--not the alphabetical characters--are used. To display or print readable text, those numeric values are then translated into characters. The translation process uses a code page, which is really nothing more than a table that equates a numeric value to a specific character. A code page may contain a

maximum of 16 rows and 16 columns of values; that is, 256 numbers representing printing and non-printing characters.

Code page switching was first introduced with DOS 3.3. With that version, DOS supports several code pages, each with an assigned number.

437	United States code page
850	Multilingual code page. Code page 850 includes all the characters used by most North American, South American, and European languages.
860	Portuguese code page.
863	French-Canadian code page.
865	Nordic code page (Norwegian and Danish languages).

The code page works in conjunction with a keyboard designation. Obviously, there must be a correlation between the code page and the keyboard used to generate the characters of a given language. The keyboards supported by DOS 3.3 are identified in Appendix B (Country and Keyboard Codes) and Appendix E (Keyboard Templates) of the *IBM Disk Operating System Version 3.3 Reference* manual.

SIDE TRIP TO INVESTIGATE THE SELECT COMMAND

Before discussing how code pages are changed, it is helpful to understand that once DOS is installed, a default country code is loaded and a default keyboard code is used. If you format a hard disk or floppy disk, you have accepted the default United States country code 001. Further, unless you specify differently, MS-DOS and PC DOS assume the use of the United State keyboard (code US).

The selection process is more direct if you use the SELECT command. It is used to assist in installing DOS on new media (floppy disk or hard disk) and to designate the keyboard layout and country code. Caution must be exercised when using the SELECT command. It is meant for use with new media only because it incorporates use of the FORMAT command.

When using the SELECT command to install DOS 3.3, you can add optional parameters to designate the country and keyboard code. Again, if you fail to designate the optional parameters, DOS assumes you wish to use the United States keyboard and country code. The syntax for the SELECT command is shown below.

```
[d:][path]SELECT s: t:[DOSpath] [cty] [Kb]
```

Because SELECT is an external DOS command, the d: and path must indicate where the SELECT command can be found. s: represents the source drive (only A: and B: are acceptable choices). t: represents the target drive, which should be either a hard disk or a floppy diskette with 720K of storage or more. In the command form shown, DOSpath is the path where DOS commands should be

installed. The cty and Kb represent, respectively, the country code and the keyboard code you wish installed.

The following command form would install from drive A: (with the SELECT command in a directory named \DOS) to drive C:. Further, it would install the DOS commands in a directory named C:\DOS and install the United States keyboard and set the country code to 001 (United States).

```
A:\DOS\SELECT A: C:\DOS 001 US
```

THE MODE COMMAND AND CODE PAGE SWITCHING

Now that you understand that SELECT installs a country code and keyboard code (whether designated or not), we can proceed with discussing the MODE command and code page switching. The MODE command can be used to specify which code page should be loaded into RAM and used. The process includes several steps because you may need to change not only the code page, but the keyboard code, too.

Figure 10-1 indicates the keyboard codes and the corresponding code pages that may be used with the keyboard code.

Country	Keyboard Code	Acceptable Code Pages
Australia	US	437, 850
Belgium	BE	437, 850
Canada (English)	US	437, 850
Canada (French)	CF	863, 850
Denmark	DK	865, 850
Finland	SU	437, 850
France	FR	437, 850
Germany	GR	437, 850
Italy	IT	437, 850
Latin America	LA	437, 850
Netherlands	NL	437, 850
Norway	NO	865, 850
Portugal	PO	860, 850
Spain	SP	437, 850
Sweden	SV	437, 850
Switzerland (French)	SF	437, 850
Switzerland (German)	SG	437, 850
United Kingdom	UK	437, 850
United States	US	437, 850

Fig. 10-1. Keyboard codes and acceptable code pages.

Switching code pages is not really a concern unless you need to create documents in a non-English language. Keep in mind the following. If you create a document after loading code page 863 (French-Canadian), and then try to read that document after loading a different code page, some of the text will be indecipherable. You must view, print, and edit a document using the same code page.

To change a code page requires several steps. Because this is not a common problem for most North American personal computer users, the steps are outlined only in general terms below.

1. You must set the COUNTRY code in the **CONFIG.SYS** file and have the COUNTRY.SYS file in your root directory.

2. Reboot your system, so it reads the CONFIG.SYS file.

> **NOTE:** The CONFIG.SYS file is only read when your system is first booted. After changing the CONFIG.SYS file, you must reboot to reset the system to the new values.

3. Execute the NLSFUNC command. This command makes sense if you understand that NLS stands for national language support and FUNC stands for function. If you fail to add parameters, NLSFUNC reads the COUNTRY.SYS file from the root directory.

4. Next, you must prepare your printer to support the appropriate code page. If you wish to prepare LPT1: for use with the French-Canadian code page, you would execute the following version of the MODE command.

```
MODE LPT1: CP PREP=863
```

5. You must then prepare the console for the same code page by entering the command

```
MODE CON: CP PREP=863
```

The two preceding commands (steps 4 and 5) merely prepare the devices (CON: and LPT1:) for code page 863. You must still use either the MODE command SELECT option or the CHCP (**CH**ange **C**ode **P**age) DOS command. If you wish to start the use of a code page for a given device, use the MODE SELECT command option. This option assumes that you wish to load the code page for a given device only; that is, not for all available devices. If you wish to use code page 863 only for the console, you would enter

```
MODE CON: CP SELECT=863
```

Execution of this command causes the printer or other attached devices to continue using the code page previously loaded.

6. To change as many attached devices to the same new code page, you would use the CHCP command. If you wish to change all devices to code page 863, you must enter

```
CHCP 863
```

The following form of this command will indicate the current code page that DOS is using:

```
CHCP
```

If you change code pages often, you probably need to enter these commands in your AUTOEXEC.BAT file or create a batch file for each code page you typically use.

Two other forms of the MODE command may be helpful in relation to code page switching. If your printer is turned off, you may lose the code page. To reinstate the code page, enter the command

```
MODE LPT1: REFRESH
```

The actual syntax of the command is

```
MODE device REFRESH
```

The device could be LPT1:, LPT2:, CON:, or any of the other reserved device names.

If you wish to know which code page(s) are currently in use for your console, use the following command:

```
MODE CON: CODEPAGE
```

The actual syntax of this command is

```
MODE device CODEPAGE
```

Once again, the device may be any parallel or serial port name, or CON: may be used for the console.

This is just a brief introduction to code page switching. Fortunately for United States and English language users, the selection of a code page, keyboard code, and country code is automatic--just use the default system choices.

There are additional details to consider when using code page switching, but because it is a rarely used or needed function, we mention only a couple of them.

The complete syntax of the MODE command (when preparing a device for code page switching) is

```
MODE device CP PREP=(list) [d:][path][codepgfile]
```

The list parameter can be one or more code page numbers. The codepgfile is a filename used to load a code page font file. When preparing the console, your options include

LCD.CPI	for use with the IBM PC Convertible LCD (liquid crystal display).
EGA.CPI	for use with all EGA (enhanced graphics adapter) type devices.

When preparing a printer, the codepgfile options include

4201.CPI	for use with the IBM 4201 Proprinter.
5202.CPI	for use with the IBM 5202 Quietwriter III printer.

Version 3.3 of IBM PC DOS supports only the above two printers when it comes to code page switching.

THE CONFIG.SYS FILE REVISITED

In earlier units, the CONFIG.SYS file was discussed. Remember that it loads certain information into memory when your computer system is booted. CONFIG.SYS is a special file that configures your system. Some of the options it provides include

1. control over the number of files open at one time,
2. control over the number of buffers used to transfer data to and from disk,
3. the ability to specify the last drive (covered in Unit 14), and
4. the ability to load device drivers.

DOS allows you to install drivers for RAM drives (RAMDRIVE.SYS or VDISK.SYS), extended memory, mice, clock/calendar cards, print spoolers, and other memory resident software. Each of these devices requires that a device driver be installed via the CONFIG.SYS file. The form of the command is

```
DEVICE=[d:][path]DRIVERNAME.EXT
```

A device driver (stored in the C:\DRIVERS directory) for an internal clock might look like the following:

```
DEVICE=C:\DRIVERS\PCCLOCK.SYS
```

One useful device driver, shipped as a separate file with PC DOS and MS-DOS, is ANSI.SYS.

SCREEN AND KEYBOARD CONTROL WITH ANSI.SYS

The ANSI.SYS driver gets its name from the American National Standards Institute. The ANSI.SYS device driver provides extended screen and keyboard support. The extended display and keyboard features provided by ANSI.SYS enable you to

1. set screen colors on a color monitor;
2. set display attributes (flashing text, high-intensity text, and inverse video) on a color or monochrome display;
3. position the cursor when at the DOS level and display messages on the screen at specified locations;
4. redefine keys on the keyboard.

To access these features, you must include the following command line in your CONFIG.SYS file:

```
DEVICE=ANSI.SYS
```

If this file is not in the root directory of your hard disk, you must specify the path to the device driver. If it resides in your \DOS subdirectory on drive C:, the command in CONFIG.SYS would read

```
DEVICE=C:\DOS\ANSI.SYS
```

COMMUNICATING WITH ANSI.SYS

Once the ANSI.SYS device driver is installed, you can communicate with the file and use its built-in features via the PROMPT command. In an earlier unit, you used the PROMPT command and a meta-string to change the DOS or system prompt. Now, with ANSI.SYS loaded, you can use the PROMPT command to change screen colors, display attributes, or the cursor location.

All ANSI.SYS sequences must begin with the <ESC> Key code. However, at the DOS prompt, pressing the <ESC> Key voids all keystrokes entered prior to pressing the key and displays a slash on the screen. Figure 10-2 displays the results of typing **`FORMAT A:`**, pressing the <ESC> Key, and then typing **`DIR A:`** before pressing the <ENTER> Key.

```
C:\>FORMAT A:\
        DIR A:

 Volume in drive A has no label
 Directory of  A:\

TOUCHY   TXT       460   5-02-89   7:18p
TECHNO   TXT      2112   5-02-89   7:00p
TRUCKIN  TXT       713   5-02-89   7:12p
DEGAS    BDC      4351   3-28-89   7:22p
DEGAS    DOC      4268   5-02-89   7:41p
EAGLE    BDC      1097   3-04-89   3:41p
EAGLE    DOC      1056   5-02-89   7:45p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
        9 File(s)    340992 bytes free

 C:\>
```

Fig. 10-2. Using the <ESC> Key to cancel the preceding portion of a command line.

Notice that the DIR command is executed, but not the FORMAT command. The point is this: simply pressing the <ESC> Key can never communicate with the ANSI.SYS driver because <ESC> serves another function at the DOS prompt.

The PROMPT command overcomes this obstacle. The $e meta-string equates with the <ESC> Key. By following the PROMPT command with the $e meta-string, you can communicate with ANSI.SYS. These sequences are referred to as "escape sequences" because they all begin with a representation of the <ESC> Key.

NOTE: You can also send the <ESC> character to activate ANSI.SYS by using some editing programs. However, the scope of this book is DOS and its uses. Therefore, we will only discuss accessing ANSI.SYS with the PROMPT command.

Further, most texts display escape sequences using the following type of notation:

```
ESC[10;15H
```

Because the PROMPT command uses the $e meta-string to represent the <ESC> Key, we use the following type of notation:

```
$e[10;15H
```

The two notations mean the same thing. In both cases an ESC or <ESC> or $e character must signal the beginning of an escape sequence.

The complete form of an escape sequence varies, depending on whether you are controlling the display color, display attributes, or cursor location. In all cases, to communicate with ANSI.SYS from the DOS prompt, the syntax includes the following:

1. PROMPT command
2. the $e meta-string
3. the left square bracket ([)
4. an appropriate numeric code or text string
5. a terminating code that indicates the specific ANSI.SYS function being executed.

The syntax would look like the following:

```
PROMPT $e[99,99z
```

In the syntax diagram 99 would be replaced by a numeric code or series of numbers (99,99) and z would be replaced by one of the following terminating codes:

A	moves the cursor up the number of rows indicated
B	moves the cursor down the number of rows indicated
C	moves the cursor forward the number of columns indicated
D	moves the cursor backward the number of columns indicated
f	moves the cursor to a specified row, column
H	moves the cursor to a specified row, column
m	modifies screen colors or display attributes
2J	clears the screen and moves the cursor to the upper left corner of the screen
K	deletes text from the cursor location to the end of the line
s	saves the current cursor location
u	restores the cursor to the last saved ([s) cursor location

NOTE: These terminating codes are case-sensitive. For instance, saving the current cursor location with the PROMPT command would require the following entry at the DOS prompt or in a batch file:

```
PROMPT $e[s
```

If you capitalize either the e or the s, the results will not be what you expect.

CURSOR MOVEMENT AND MESSAGE DISPLAYS WITH ANSI.SYS

The PROMPT command can also be used to move the cursor to specific screen locations and display messages. The terminating codes used for cursor movement were listed earlier. The exact syntax of each of the escape sequences is shown below.

`PROMPT $e[row;columnH`	Moves the cursor to the row and column indicated.
`PROMPT $e[row;columnf`	Moves the cursor to the row and column indicated.
`PROMPT $e[rowA`	Moves the cursor up the number of rows indicated.
`PROMPT $e[rowB`	Moves the cursor down the number of rows indicated.
`PROMPT $e[columnC`	Moves the cursor forward the number of columns indicated.

`PROMPT $e[columnD`	Moves the cursor backward the number of columns indicated.
`PROMPT $e[s`	Saves the current cursor location.
`PROMPT $e[u`	Restores the cursor to the location last saved using the ESC[[e sequence.
`PROMPT $e[2J`	Moves the cursor to the upper left corner--the home position; same function as performed by the CLS command.
`PROMPT $e[K`	Deletes text from the current cursor location to the end of the current line.

A typical screen displays 80 characters horizontally across each line and 25 lines vertically on the screen. Some screens can display more or fewer characters across (40 or 132) and more lines per screen (43 or 50). Each of the 25 or more vertical lines is considered a row and numbered 1 to 25. Each character position (beginning at the left edge of the screen) is considered a column and numbered 1 to 80. Therefore, to move to the dead center of a typical 80-column by 25-line screen, you would enter

```
PROMPT $e[13;40f
```

or

```
PROMPT $e[13;40H
```

To move the cursor down five rows after the last cursor location, you would use the command form

```
PROMPT $e[5B
```

NOTE: If you use the PROMPT command to access ANSI.SYS and wish it to also display the default drive and path, you must include the PROMPT pg command in your batch file or reexecute it at the DOS prompt. Using PROMPT to access ANSI.SYS negates any other previous use of PROMPT.

GUIDED ACTIVITY

USE THE PROMPT COMMAND AND ANSI.SYS TO MOVE THE CURSOR: In this Guided Activity, you will use the PROMPT command to communicate with the ANSI.SYS driver and modify the location and display of your system prompt.

1. Turn on your computer system, and go to the C:\DOS directory or the directory containing your DOS command files.

2. At the DOS prompt, type **CLS** and press <ENTER>.

3. Type **PROMPT $e[5B** and press <ENTER>.

Notice that the DOS prompt disappears and the cursor drops down five rows (lines).

4. Type **CLS** and press <ENTER>. Because the PROMPT command is set to drop five rows, the cursor cannot go to the home position--as it normally would when invoking the CLS command.

5. Type **JUNK1** and press <ENTER>.

6. Type **JUNK2** and press <ENTER>. Each time, the cursor drops another five lines.

7. Type **JUNK3** and press <ENTER>. Again, the cursor drops another five lines.

8. Type **JUNK4** and press <ENTER>. Because the PROMPT command (accessing ANSI.SYS) cannot move the cursor off the screen, the cursor only drops a couple lines--even though five were indicated.

Figure 5-3 shows the results of the last few steps.

```
JUNK1
Bad command or file name



JUNK2
Bad command or file name



JUNK3
Bad command or file name

JUNK4
Bad command or file name
```

Fig. 10-3. Using the PROMPT command to drop five lines.

9. Type **CLS** and press <ENTER>. Only the cursor remains on the sixth line down. CLS moved the cursor to row 1, and the PROMPT command moved it down five more lines.

The next step will cause the DOS prompt to reappear, displaying the current drive and path.

10. Type **PROMPT pg** and press <ENTER>.

11. Type **CLS** and press <ENTER>. Because the prompt was reset in step 10, the cursor can once again move to the home position on the screen.

In the next step, you combine three meta-strings in order to move the cursor (while still displaying the DOS prompt indicating the default drive and path).

12. Type **PROMPT $e[5B pg** and press <ENTER>.

The results of the previous step are displayed in Fig. 10-4.

```
C:\DOS>PROMPT $e[5B $p$g



 C:\DOS>
```

Fig. 10-4. Combining three PROMPT command meta-strings.

The next meta-string will cause the DOS prompt to erase the entire screen, move to the home position, and then display the DOS prompt with the default drive and path.

13. Type **`PROMPT $e[2J $p$g`** and press <ENTER>.

14. Press <ENTER> several times. Notice how the prompt behaves. It continues to clear the screen and stay in the upper left corner of the screen.

15. Type **`DIR`** and press <ENTER>. What happens at the end of the DIR listing? Do you understand why?

In the next PROMPT command form, you will store the current cursor location, place the current time at the top of the screen (home position), return to the stored cursor location, and then display the DOS prompt with the default drive and path.

16. Type **`PROMPT $e[s $e[1;1H $t $e[u $p$g`** and press <ENTER>.

Before proceeding, let's dissect this command form:

`$e[s`	Saves the current cursor location.
`$e[1;1H`	Moves the cursor to row 1, column 1.
`$t`	Displays the current system time.
`$e[u`	Moves the cursor to the saved cursor location.
`$p$g`	Displays the DOS prompt with the default drive and path.

17. Type **DIR** and press <ENTER>.

Notice the leftover detail on the top line along with the system time. The next command form includes the $e[K meta-string. It clears text from the cursor location to the end of the current line and causes the time to appear on the top line by itself.

18. Type **PROMPT $e[s $e[1;1H $t $e[K $e[u pg** and press <ENTER>.

The line with the system time is now cleared.

19. Press <ENTER> two or three times. The cursor remains at the bottom of the screen, and the time remains by itself on the top line.

20. Type **DIR** and press <ENTER>.

21. Type **CLS** and press <ENTER>. The time continues to change as the system time changes.

In the next step, two more meta-strings are invoked.

`$e[25C`	Moves the cursor 25 columns (characters) forward on the line.
`$dc`	Displays the system date.

22. Type **PROMPT $e[s $e[1;1H $t $e[K $e[25C $d $e[u pg** and press <ENTER>.

23. Press <ENTER> several times, and notice the date and time displayed at the top of the screen.

In Fig. 10-5, the PROMPT command displays the system date and time at the top of the monitor screen.

```
13:29:25.07                                        Tue  5-30-1989
C:\DOS>PROMPT $e[s $e[1;1H $t $e[K $e[25C $d $e[u $p$g

C:\DOS>
C:\DOS>
C:\DOS>
C:\DOS>
```

Fig. 10-5. Date and time displayed using the DOS PROMPT command.

In order to return your system to normal, complete the next step.

24. Type **PROMPT pg** and press <ENTER>.

25. Type **CLS** and press <ENTER>.

> **NOTE:** It can get very tedious typing these long PROMPT commands with meta-strings. To save time and effort, you can enter these commands from either an AUTOEXEC.BAT file or any batch file.

SETTING SCREEN COLORS AND DISPLAY ATTRIBUTES WITH ANSI.SYS

The following codes are used in escape sequences to establish the colors and display attributes listed. Note that some numbers are specifically used for background colors while others are used for foreground (text) colors. Still other codes are used to generate display attributes.

0	All attributes are turned off
1	High-intensity or bold text
4	Underscored or underlined text (only on a monochrome display)
5	Blinking or flashing text
7	Inverse video
8	Hidden or concealed text
30	Foreground Black
31	Foreground Red
32	Foreground Green
33	Foreground Yellow
34	Foreground Blue
35	Foreground Magenta
36	Foreground Cyan
37	Foreground White
40	Background Black
41	Background Red
42	Background Green
43	Background Yellow
44	Background Blue
45	Background Magenta
46	Background Cyan
47	Background White

Remember that when setting screen colors or display attributes, you must use the m terminating code for your escape sequence. Keeping these guidelines and codes in mind, you could change your color monitor's background screen color to blue with the command form

```
PROMPT $e[44m
```

To change the background color to green, you would use the command form

```
PROMPT $e[42m
```

The foreground or text color can be altered by inserting a number corresponding to the foreground color desired. Magenta text can be generated by the command form

```
PROMPT $e[35m
```

> **NOTE:** If you set the background and foreground to the same color, you will not be able to read the screen text.

Both the foreground and the background colors may be set with a single command line. To establish white text on a green background, you would use the command form

```
PROMPT $e[42;37m
```

or

```
PROMPT $e[37;42m
```

Finally, you could establish a text display attribute in the same command line by adding the appropriate code. The following command form would cause flashing white text on a blue background:

```
PROMPT $e[5;37;44m
```

GUIDED ACTIVITY

USE THE PROMPT COMMAND TO ALTER SCREEN COLORS AND ATTRIBUTES: The first portion of this Guided Activity can be accomplished on any system with ANSI.SYS loaded. The last portion requires a color monitor.

1. Turn on your computer system, and go to the C:\DOS directory or the directory containing your DOS command files.

2. At the DOS prompt, type **CLS** and press <ENTER>.

3. Type **PROMPT $e[5m** and press <ENTER>.

The system prompt disappears. You have invoked a display attribute meta-string that will cause the text on the screen to blink. However, nothing is blinking or flashing until you type some characters.

4. Type **DIR** and press <ENTER>. As the listing scrolls down the screen, it will be flashing.

5. Before the display drives you crazy, type **PROMPT $e[0m pg** and press <ENTER>.

This command form turns off the blinking for any new text and displays the DOS prompt with the default drive and path. Notice the prompt is not blinking, but the residual text is.

6. Type **CLS** and press <ENTER>.

In the next step you will add a screen heading in bold or high-intensity type. The bold will be turned on, then off and the cursor moved to row 3, column 1 before the DOS prompt is displayed.

7. Type `PROMPT $e[H $e[1m MY COMPUTER SYSTEM $e[0m $e[3;1H$p$g` and press <ENTER>.

Before proceeding, let's examine the several meta-strings composing this command.

`$e[H`	Moves the cursor to the home position; when row and column numbers are omitted, DOS assumes the home position as the target.
`$e[1m`	Turns on bold text.
`MY COMPUTER SYSTEM`	Is the text string to appear in bold text.
`$e[0m`	Turns off all display attributes; that is, returns the display to normal.
`$e[3;1H$p$g`	Moves the cursor to row 3, column 1 and displays the system prompt--without displaying a space first.

Figure 10-6 displays the execution of this command.

```
  MY COMPUTER SYSTEM
C:\DOS>PROMPT $e[H $e[1m MY COMPUTER SYSTEM $e[0m $e[3;1H$p$g
C:\DOS>
```

Fig. 10-6. Bold attribute screen text generated by PROMPT and ANSI.SYS.

8. Type **CLS** and press <ENTER>.

9. Type **DIR** and press <ENTER>.

The prompt got buried in the directory listing. This would not happen if we cleared the line of all text after the DOS prompt. In the next step you will clear the first line, center the MY COMPUTER SYSTEM text string, and display the default drive and path through the DOS prompt.

10. Type **CLS** and press <ENTER>. This clears the screen to help you see what to enter next.

11. Type **PROMPT $e[H $e[K $e[1m $e[31C MY COMPUTER SYSTEM $e[0m $e[2;1H $e[K $e[3;1H $e[K$p$g** and press <ENTER>.

There are four additional meta-strings in this command.

`$e[31C`	Moves the cursor over 31 columns before displaying the text string MY COMPUTER SYSTEM.
`$e[2;1H`	Moves the cursor to row 2, column 1.
`$e[K`	Clears text from the cursor location to the end of the current line.
`$e[Kp$g`	Clears the line and then displays the system prompt.

The screen still looks a bit messy but can be cured by invoking the CLS command.

12. Type **CLS** and press <ENTER>.

> **NOTE:** The following steps require use of a color monitor.

13. Type **PROMPT $e[44;33m pg** and press <ENTER>.

By adding the pg meta-string, we preserve the DOS prompt. The only change appears to be the color change of the block behind the DOS prompt. Though the codes indicate a blue background color and a yellow foreground color, the yellow may look more brown or tan, depending on how you have adjusted your monitor's brightness control.

14. Press <ENTER> three or four times. Notice the spreading of the new background and foreground colors.

15. Type **DIR** and watch the screen as you press <ENTER>. The new colors continue to spread.

16. Type **PROMPT $e[34;43m pg** and press <ENTER> to reverse the colors.

17. Type **DIR** and watch the screen as you press <ENTER>.

In the next step, you will only change the foreground (text) color to white.

18. Type **PROMPT $e[37m pg** and press <ENTER>.

19. Type **CLS** and press <ENTER>. The screen goes black except for the area around the DOS prompt.

20. Type **DIR** and press <ENTER>.

The following step will change only the background color from yellow to blue.

21. Type **PROMPT $e[44m pg** and press <ENTER>.

22. Type **TREE** and watch your screen as you press <ENTER>.

The following command restores your display attributes and colors to normal.

23. Type **PROMPT $e[40;37m pg** and press <ENTER>.

24. Type **CLS** and press <ENTER>.

KEYBOARD REASSIGNMENT WITH ANSI.SYS

The final use of the PROMPT command is to communicate with ANSI.SYS in order to redefine keyboard keys. This may not seem very functional. After all, why would you ever want to redefine a J to generate an O or a D to generate an I? DOS can access the extended ASCII character set which includes characters not available on the typical computer keyboard, such as the ¼ and ½ characters.

To reassign a key, you would use the following PROMPT command syntax.

```
PROMPT $e[oldkey;newkeyp:
```

You can also redefine a key to generate a text string, by replacing the newkey parameter with a text string.

Unfortunately, it is not as simple as entering the old and new key names. You must use the scan codes (numbers) for the old and new keys. The p is the required termination code for this function. If you want the key redefinition to conclude with a carriage return, the termination code must be 13p rather than just 13.

The following table lists the scan codes for the IBM PC. Notice that codes are given for the key by itself and in combination with the <SHIFT>, <CTRL>, and <ALT> Keys.

Key	Key Code	With <SHIFT>	With <CTRL>	With <ALT>
a	97	65	1	0;30
b	98	66	2	0;48
c	99	67	3	0;46
d	100	68	4	0;32
e	101	69	5	0;18
f	102	70	6	0;33
g	103	71	7	0;34
h	104	72	8	0;35
i	105	73	9	0;23
j	106	74	10	0;36
k	107	75	11	0;37
l	108	76	12	0;38
m	109	77	13	0;50
n	110	78	14	0;49
o	111	79	15	0;24
p	112	80	16	0;25
q	113	81	17	0;16
r	114	82	18	0;19
s	115	83	19	0;31
t	116	84	20	0;20
u	117	85	21	0;22
v	118	86	22	0;47
w	119	87	23	0;17
x	120	88	24	0;45
y	121	89	25	0;21
z	122	90	26	0;44
1	49	63		0;120
2	50	64		0;121
3	51	35		0;122
4	52	36		0;123
5	53	37		0;124
6	54	94		0;125
7	55	38		0;126
8	56	42		0;127
9	57	40		0;128
0	48	41		0;129
-	45	95		0;130
=	61	43		0;131
TAB	9	0;15		
SPACE	32			

The following are scan codes for characters not on the standard IBM keyboard:

¢	155
¿	168
½	171
¼	172
¡	173
«	174
»	175
alpha	224
beta	225
gamma	226
pi	227
sigma	228
tau	231
theta	233
omega	234
phi	237

NOTE: These codes are identical to the ASCII codes that are numeric representations of the characters, numbers, and symbols generated by pressing keys on the keyboard. The entire list has not been included, just the most commonly used character values.

To reassign the <ALT-2> Key combination to generate a ½ symbol, you would use the following form of the PROMPT command:

```
PROMPT $e[0;121;171p
```

Additionally, if you wanted <ALT-4> to generate a ¼ symbol, the command form would be

```
PROMPT $e[0;123;172p
```

If you want to cause the <CTRL-D> Key combination to generate a DIR listing of drive A:, you can use the following command form:

```
PROMPT $e[4;"DIR A:"13p
```

Notice that the multiple keystrokes are enclosed in quotation marks and that the 13p ends the string with a carriage return. After redefining this key, you can simple press <CTRL-D> to generate a DIR listing from drive A:.

NOTE: To reset a key to its original definition, use the following syntax:

```
PROMPT $e[oldkey;oldkeyp
```

If you had redefined the letter A (capital A), you can reset the key to its original use by entering

```
PROMPT $e[65;65p
```

GUIDED ACTIVITY

REDEFINE KEYBOARD KEYS WITH ANSI.SYS AND THE PROMPT COMMAND: In this Guided Activity, you will redefine a couple of keys using the PROMPT command to access the redefinition features of ANSI.SYS.

1. Turn on your computer system, and go to the C:\DOS directory or the directory containing your DOS command files.

2. At the DOS prompt, type **CLS** and press <ENTER>.

3. Type **PROMPT $e[32;49p pg** and press <ENTER>.

The first meta-string redefines the <SPACEBAR> (code 32) to be a number one (code 49). The p is the terminating code. Now when you press the <SPACEBAR>, you will generate a "1".

4. Press <SPACEBAR> three times, and press <ENTER>.

```
C:\DOS>PROMPT $e[32;49p $p$g

C:\DOS>111
Bad command or file name

C:\DOS>
```

Fig. 10-7. The redefined <SPACEBAR> Key.

5. Press <SPACEBAR>, and then press <ENTER>. You'll notice that the <SPACEBAR> is still redefined.

In the next step, you will attempt to redefine the <SPACEBAR> back to its original use.

6. Type `PROMPT $e[32;32p $p$g` and press <ENTER>.

Notice the "Bad command or file name" error message, as shown in Fig. 10-8. If you look at your screen and Fig. 10-8 carefully, you will see that when you attempted to type the space after PROMPT, it came out "PROMPT1". You have redefined the <SPACEBAR> and actually have no way to redefine it. Whenever redefining keys, be sure you do not "paint yourself into a corner".

```
C:\DOS>PROMPT $e[32;49p $p$g

C:\DOS>111
Bad command or file name

C:\DOS>1
Bad command or file name

C:\DOS>PROMPT1$e[32;4913p1$p$g
Bad command or file name

C:\DOS>
```

Fig. 10-8. Screen showing the redefined <SPACEBAR> character as a "1".

The only way to regain the space character is to reboot your system.

7. Press <CTRL>, <ALT>, and <DEL> simultaneously to reboot your system.

8. If prompted, enter the date and time.

9. Type **CLS** and press <ENTER>.

In the next step, you will redefine the <CTRL-D> Key combination to generate a text string to invoke the `DIR A:` command.

10. Type **`PROMPT $e[4;"DIR A:";13p $p$g`** and press <ENTER>.

Notice that the 13p terminating code means that <CTRL-D> will generate the text string DIR A: and follow it with a carriage return (the character generated by pressing <ENTER>).

11. Place the West Student Data Disk in drive A:.

12. Press <CTRL-D>. You do NOT need to follow this by pressing the <ENTER> Key.

Figure 10-9 displays the results of pressing <CTRL-D>. Notice that the results were the same as if you had typed `DIR A:` and pressed <ENTER> at the DOS prompt.

```
C:\>PROMPT $e[4;"DIR A:";13p $p$g

C:\>DIR A:

 Volume in drive A has no label
 Directory of  A:\

TOUCHY   TXT      460   5-02-89   7:18p
TECHNO   TXT     2112   5-02-89   7:00p
TRUCKIN  TXT      713   5-02-89   7:12p
DEGAS    BDC     4351   3-28-89   7:22p
DEGAS    DOC     4268   5-02-89   7:41p
EAGLE    BDC     1097   3-04-89   3:41p
EAGLE    DOC     1056   5-02-89   7:45p
OVERDUE  DOC      617   3-04-89  10:17a
PCAT     DOC      679   4-22-89   7:41p
        9 File(s)    340992 bytes free

 C:\>
```

Fig. 10-9. Keystrokes generated by pressing <CTRL-D>.

The final step is to redefine the <CTRL-D> Key combination so that it no longer generates the text and carriage return.

13. Type `PROMPT $e[4;4p $p$g` and press <ENTER>.

By listing the code (4) for <CTRL-D> twice, you in effect tell DOS and ANSI.SYS to let those keystrokes represent their normal function.

14. Press <CTRL-D>, and press <ENTER>.

This time the screen simply shows the ^D characters, as displayed in Fig. 10-10.

```
C:\>PROMPT $e[4;"DIR A:";13p $p$g

C:\>DIR A:

 Volume in drive A has no label
 Directory of  A:\

TOUCHY   TXT       460   5-02-89   7:18p
TECHNO   TXT      2112   5-02-89   7:00p
TRUCKIN  TXT       713   5-02-89   7:12p
DEGAS    BDC      4351   3-28-89   7:22p
DEGAS    DOC      4268   5-02-89   7:41p
EAGLE    BDC      1097   3-04-89   3:41p
EAGLE    DOC      1056   5-02-89   7:45p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
        9 File(s)    340992 bytes free

 C:\>PROMPT $e[4;4p $p$g

 C:\>^D
Bad command or file name

 C:\>
```

Fig. 10-10. Normal screen display generated by pressing <CTRL-D>.

COMMAND REVIEW DICTIONARY

In this unit you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

CHCP	Indicates the current code pages used by DOS.
DEVICE=ANSI.SYS	Installs the ANSI.SYS device driver when included in the CONFIG.SYS file.
MODE BW40	Causes a color monitor to display a maximum of 40 characters per line across the screen in monochrome.
MODE BW80	Causes a color monitor to display a maximum of 80 characters per line across the screen in monochrome.
MODE CO40	Causes a color monitor to display a maximum of 40 characters per line across the screen in color.
MODE CO80	Causes a color monitor to display a maximum of 80 characters per line across the screen in color.
MODE CON: CODEPAGE	Indicates current code page(s) in use for your console.
MODE MONO	Causes the monochrome display adapter to be the active display adapter and generates 80 characters per line across the screen.
MODE 40	Causes a color monitor to display a maximum of 40 characters per line across the screen.
MODE 80	Causes a color monitor to display a maximum of 80 characters per line across the screen.
PROMPT $e[40;37m$p$g	Resets the monitor's display to a black background color and a white foreground (text) color; also displays the default drive and path.

REVIEW QUESTIONS

1. What are the five uses of the MODE command?
2. What action does the `MODE CO80` command invoke?
3. What action does the `MODE MONO` command invoke?
4. What action does the `MODE BW40` command invoke?
5. What action does the `MODE MONO,L,T` command invoke?
6. Why are code pages used?
7. How many code pages are supported by DOS 3.3?
8. When using the SELECT command, which are the default keyboard and country codes?
9. Why is the SELECT command used only when setting up new media?
10. Which code page may be used with all keyboard codes?
11. How do you load the ANSI.SYS driver?
12. What are the four uses of the CONFIG.SYS file?
13. What does ANSI stand for?
14. Which DOS command is used to communicate with the ANSI.SYS device drive?
15. What does the $e meta-string represent?
16. What always ends an escape sequence?
17. What action occurs when PROMPT $e[s is executed?
18. Which version of the PROMPT command deletes text from the cursor location to the end of the current line?
19. Which version of the PROMPT command displays the default drive and path as part of the DOS prompt?
20. How are the results of the following two commands different?

```
PROMPT $e[42;37m
PROMPT $e[37;42m
```

21. Can the PROMPT command be used to redefine the letter A to display a letter Z on the monitor screen?

22. Are the scan codes for the IBM PC keyboard the same as the ASCII codes for the extended character set?

23. What would happen if you redefined the <SPACEBAR> to print the letter G?

DOCUMENTATION RESEARCH

Check Chapter 7 of the *IBM Disk Operating System Version 3.30 Reference* manual or Chapter 3 of the *Microsoft MS-DOS User's Reference* to answer these questions.

1. List the abbreviations for the following MODE command CODEPAGE options:

   CODEPAGE
   PREPARE
   REFRESH
   SELECT
   STATUS

2. Does use of the MODE command increase the resident size of DOS?*

3. Which type of file has a .CPI extension?

* Not covered in the MS-DOS documentation.

APPLICATION H

PRINTER CONTROL COMMANDS

PART 1: CREATING AN ANSI.SYS SCREEN CHANGE BATCH FILE

In Unit 10, you learned about the ANSI.SYS device driver and how the PROMPT command can communicate with it to modify your screen and keyboard. In this application, you will create some batch files to make it easier to change your screen colors and move the cursor.

A. Create a batch file named COLORS.BAT that cycles through five different foreground and background color combinations. Be sure that the entire screen is displayed in the new text and background colors, that the batch file stops until a key is pressed, and that the file then moves to the next color combination. The final color combination should be a black background with a white foreground color.

B. Modify the above file to include a saying centered in the middle of your screen (one saying for each color combination). The sayings may be poems, jokes, or statements of fact--you choose. Save this file as COLORS2.BAT.

C. Modify COLORS2.BAT to display the current date and time on the top line of your screen. Save this file as COLORS3.BAT.

D. Turn in a disk with all three files (COLORS.BAT, COLORS2.BAT, and COLORS3.BAT) to your instructor.

UNIT 11

PRINTER CONTROL COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. list and explain the forms of the MODE command that provide printer control;
2. create a PRINT command queue;
3. define a print queue;
4. describe the benefits of a print queue;
5. list the most common parameters used with the PRINT command.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. MODE
2. PRINT.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): background task, foreground task, print queue, and time-out.
____Guided Activity: Use the Print MODE Option.
____Guided Activity: Create a Print Queue.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

THE MODE COMMAND--CHANGING PRINTER OUTPUT

Printer output can be affected by using the following MODE command syntax:

```
MODE LPTn[:][c][,l][,P]
```

where n represents the printer port number (that is, LPT1, LPT2, or LPT3), c represents the characters per line, and l is the lines per vertical inch. The P parameter indicates that you want the system to continuously retry the serial port should a **time-out** error occur. A time-out would occur should the printer be off-line, turned off, in an error situation (ribbon out or paper jammed), or busy. The latter situation might occur if the printer is connected to a network system or if it is shared by two or more printers via an electronic autoscan switch.

The power-on default value for this MODE command option is

```
MODE LPT1:80,6
```

The options are limited. For vertical lines per inch, only two values may be used: 6 or 8. Similarly, only the values 80 or 132 may be used for characters per line. It should be noted that these commands do not work with all printers. A prime example is the Hewlett Packard LaserJet Series II. You cannot adjust the lines per vertical inch or the characters per line using the DOS MODE command. This is true of any printer with an 8-in. platen that does not support 132 columns of printable characters.

THE MODE COMMAND--SWITCHING FROM PARALLEL TO SERIAL

Another optional use of the MODE command is to switch output from a parallel port (LPT1:, LPT2:, or LPT3:) to a serial port (COM1:, COM2:, COM3:, or COM4:). The command syntax is

```
MODE LPTn[:]=COMm[:]
```

where n is the number of the parallel port and m represents the serial port number. Remember, the parallel port options are only 1, 2, or 3, while the serial port options are 1, 2, 3, or 4.

To send data output to serial port 1 instead of parallel port 1, the command form would be

```
MODE LPT1:=COM1:
```

DOS normally sends data to your printer through a parallel port. However, many printers are designed to attach to a serial port. If you have a serial printer connected to your system, you would use the above command to redirect output from parallel port one (LPT1:) to serial port one (COM1:). Of course, you could also redirect the output to a different serial port. The following MODE command form redirects output from LPT1: to COM3::

```
MODE LPT1:=COM3:
```

To cancel the redirection, you would simply enter

```
MODE LPT1:
```

GUIDED ACTIVITY

USE THE PRINT MODE OPTION: All the MODE command forms are hardware-dependent, so it is difficult to present Guided Activities to demonstrate each option. This activity assumes that a dot-matrix printer is attached to your microcomputer. First, you will set the printer output to condensed print (132 characters per line, 8 lines per vertical inch) and print a directory listing. Then the activity will guide you in resetting the printer output to 80 characters per line with 6 lines per vertical inch.

1. Log on to the hard-disk directory containing your DOS files, probably C:\DOS.

If you do not have a printer attached to parallel port one, do not proceed. Rather, substitute a port designation to which a printer is attached. If no printer is available, do not complete this exercise.

2. Be sure the printer is turned on and the paper is set to the top of the form.

3. Type **MODE LPT1:132,8** and press <ENTER>.

The screen will return the following message:

LPT1: set for 132
No retry on parallel printer time-out

4. Type **DIR > LPT1:** and press <ENTER>.

Check your printer and notice the condensed print size.

5. Type **MODE LPT1:80,6** and press <ENTER>.

The monitor screen will respond with the message

LPT1: set for 80
No retry on parallel printer time-out

6. Type **DIR > LPT1:**

7. Press <ENTER>.

Check your printer and notice that the print size has returned to normal; that is, 10 characters per inch.

BACKGROUND PRINTING WITH THE PRINT COMMAND

Printing a long document can tie up your personal computer for several minutes, wasting your time until the tedious task is complete. Without special software or the PRINT command, you can get very frustrated. The PRINT command, which first became available with DOS 2.x, helps relieve this problem.

The PRINT command creates a queue, or storage area for files to be printed to a specified printer. With the introduction of the PRINT command, PC DOS took a step toward multitasking; that is, handling more than one task at a time.

The PRINT command syntax, in its simplest form, is

```
PRINT [d:][filename.ext]
```

The PRINT command has several optional parameters. The most common are listed below. The first six parameters are used only the first time the PRINT command is invoked in each computing session.

/D:device	Indicates to which serial or parallel port the printer is connected. The default device is PRN; that is, LPT1:. This parameter, if used, must always be the first parameter following the word PRINT.

/B:buffer size — Indicates the reserved memory (in bytes) for the PRINT command queue. The default size is 512 bytes. The maximum size of the print buffer is 16,386 bytes (16K).

/Q:queue-size — Specifies the number of files that may be queued at one time. The range of values for MS-DOS is 4 to 32 files, with the default value being 10 files. PC DOS supports a range of 1 to 32 files.

/U: — Specifies the clock ticks the PRINT command will wait for the printer. The default is 1. If the wait is longer than the specified clock tick, the PRINT command gives up its slice of CPU time.

/M: — Specifies the clock ticks that the PRINT command can use to send characters to the printer. The default is 2, but the range includes values from 1 to 255.

/S: — Indicates the time slice interval used by the DOS scheduler for the PRINT command. The default slice is 8.

/T — Terminates the queue; that is, cancels all files in the queue and stops printing. When printing is stopped, paper in the printer is advanced to the next top of the form.

/C — Cancels selected files. The /C parameter is tacked onto the end of the filename. All files listed on the command line after this parameter are also cancelled until a /P is encountered.

/P — Prints selected files and/or queues them for printing. The /P parameter is tacked onto the end of a filename on the command line. All files listed on the command line after this parameter are also queued for printing until a /C is encountered.

Upon booting your system, you would use the following command form to create a PRINT command queue to the serial port 1 and reserve 5120 bytes (5K) of memory for a queue that can handle five files:

```
PRINT /D:COM1:/B:5120/Q:5
```

The same size buffer for parallel port 1 would be generated by

```
PRINT /D:PRN/B:5120/Q:5
```

or

```
PRINT /B:5120/Q:5
```

If the second form is used (with the device statement omitted), DOS will prompt you for the "Name of list device". If you press the <ENTER> Key, the default setting of PRN or LPT1: will be assumed by DOS.

If three files named LETTER1.DOC, LETTER2.DOC, and LETTER3.DOC were to be printed, either of two command forms could be used. The first would list multiple filenames after the PRINT command:

```
PRINT LETTER1.DOC LETTER2.DOC LETTER3.DOC
```

The other option would use global wildcard characters:

```
PRINT LETTER?.DOC
```

The /P parameter is assumed unless it follows a /C parameter on the same command line. In the above example, if a file LETTER4.DOC existed on the disk and should not be printed, the command could be modified to read

```
PRINT LETTER?.DOC LETTER4.DOC/C
```

Immediately after entering the above command, the monitor provides a listing of active files in the queue.

The listing of files in the queue can be displayed at any time by entering

```
PRINT
```

The screen response in Fig. 11-1 shows the response to both command forms.

```
C:\>PRINT LETTER?.DOC LETTER4.DOC/C

   C:\LETTER1.DOC is currently being printed
   C:\LETTER2.DOC is in queue
   C:\LETTER3.DOC is in queue

C:\>PRINT

   C:\LETTER1.DOC is currently being printed
   C:\LETTER2.DOC is in queue
   C:\LETTER3.DOC is in queue

C:\>
```

Fig. 11-1. Screen response indicating the status of the queue.

All files in the queue, including the file currently being printed, could be cancelled by entering the command form

```
PRINT /T
```

There are several commands that can interfere with the PRINT command queue. They include any attempt to use the printer, such as redirecting the output of a DOS command to the printer port used by the PRINT command queue, or trying to use the PrtSc (Print Screen) Key.

It is also important to avoid altering files in the queue until they have been completely printed and released from the queue.

Most word processors provide the capability of printing document files to disk, rather than directly to the printer. The most effective use of the PRINT command is to print files to disk, thus saving any control characters used to format printed output. Later the PRINT command can be used to queue several files for printing. While those files are printing, you can return to other computing tasks.

Some DOS commands slow down the printing process and result in an appreciable slowdown in either the printing or the computing task. This is because in a multitasking environment, one task is given priority over the other. The high-priority task runs in the foreground and is called the **foreground task**, while the low-priority task runs in the background and is called the **background task**.

We began this discussion by indicating that the PRINT command operates in the background because use of the command frees you to return to another application. Thus, it appears that the PRINT command operates in the background, while you tackle another task in the foreground. Technically, the reverse is true. The PRINT command is really the high-priority task and runs in the foreground. More important than which task runs in the foreground is the fact that at least the PRINT command is a step in the right direction--that of making PC DOS and MS-DOS computers multitasking machines.

GUIDED ACTIVITY

CREATE A PRINT QUEUE: In this Guided Activity, you will print several text files from the West Student Data Disk. You will use the PRINT command to send the files to a print queue for printing.

1. Change directories so you are in the root directory of your hard disk.

Because you may need to modify your AUTOEXEC.BAT file, you will first make a backup of it under the title AUTOEXEC.OLD.

2. Type **COPY AUTOEXEC.BAT AUTOEXEC.OLD** and press <ENTER>.

3. Type **EDLIN AUTOEXEC.BAT** and press <ENTER>.

4. Type **L** and press <ENTER>.

5. Check the file for a PRINT command. Most likely it will look like

```
PRINT /D:LPT1
```

6. If you find such a line, delete it.

7. Check also to see if a PATH command exists. It probably reads

```
PATH=C:\DOS
```

8. If you do not find it, add the command.

9. Type **E** and press <ENTER> to exit EDLIN.

10. Reboot your system, and enter the current date and time when prompted.

11. Insert the West Student Data Disk in drive A:.

12. Type **A:** and press <ENTER>.

13. Type **DIR** and press <ENTER>.

The screen response should appear as shown in Fig. 11-2.

```
C:\>A:

A:\>DIR

 Volume in drive A has no label
 Directory of  A:\

TOUCHY   TXT       460   5-02-89   7:18p
TECHNO   TXT      2112   5-02-89   7:00p
TRUCKIN  TXT       713   5-02-89   7:12p
DEGAS    BDC      4351   3-28-89   7:22p
DEGAS    DOC      4268   5-02-89   7:41p
EAGLE    BDC      1097   3-04-89   3:41p
EAGLE    DOC      1056   5-02-89   7:45p
OVERDUE  DOC       617   3-04-89  10:17a
PCAT     DOC       679   4-22-89   7:41p
        9 File(s)    340992 bytes free

A:\>
```

Fig. 11-2. DIR listing of copied files.

14. Be sure your printer is turned on and the paper is set to the top of the form.

Next, we must set up the **print queue**.

15. Type **CLS** and press <ENTER>.

16. Type **PRINT /D:LPT1** and press <ENTER>. This form of the command indicates that data sent to the printer from the print queue will be directed to LPT1:. This command uses the default PRINT command settings.

The screen display will appear as shown in Fig. 11-3. This command assumes that the queue will use the default values; that is, 10 files maximum in the queue and 512 bytes reserved for the queue.

```
A:\>PRINT /D:LPT1
Resident part of PRINT installed
PRINT queue is empty

A:\>
```

Fig. 11-3. Screen response to setup of PRINT command queue.

Now you will send all files on drive A: to the queue.

17. Type **PRINT *.*** and press <ENTER>.

Your printer will begin printing out the text of the first file, and the monitor will respond with the message displayed in Fig. 11-4.

```
A:\>PRINT /D:LPT1
Resident part of PRINT installed
PRINT queue is empty

A:\>PRINT *.*

    A:\TOUCHY.TXT is currently being printed
    A:\TECHNO.TXT is in queue
    A:\TRUCKIN.TXT is in queue
    A:\DEGAS.BDC is in queue
    A:\DEGAS.DOC is in queue
    A:\EAGLE.BDC is in queue
    A:\EAGLE.DOC is in queue
    A:\OVERDUE.DOC is in queue
    A:\PCAT.DOC is in queue

A:\>
```

Fig. 11-4. Status report on the print queue.

18. Quickly type **PRINT *.DOC/C** and press <ENTER> to cancel the files ending with the .DOC extension.

The monitor screen will display a message indicating the status of files in the print queue. The files ending with .DOC should not appear, because the last command removed them from the queue.

The message will look similar to that shown in Fig. 11-5, depending on how quickly the last command was executed.

```
A:\>PRINT /D:LPT1
Resident part of PRINT installed
PRINT queue is empty

A:\>PRINT *.*

    A:\TOUCHY.TXT is currently being printed
    A:\TECHNO.TXT is in queue
    A:\TRUCKIN.TXT is in queue
    A:\DEGAS.BDC is in queue
    A:\DEGAS.DOC is in queue
    A:\EAGLE.BDC is in queue
    A:\EAGLE.DOC is in queue
    A:\OVERDUE.DOC is in queue
    A:\PCAT.DOC is in queue

A:\>PRINT *.DOC/C

    A:\DEGAS.BDC is currently being printed
    A:\EAGLE.BDC is in queue

A:\>
```

Fig. 11-5. Status report on print queue after removing .DOC files.

19. If the printer is still running, type **PRINT /T** and press <ENTER> to terminate all files in the queue.

The printing should cease, and the screen will indicate that

PRINT queue is empty

> **NOTE:** If you did not modify your AUTOEXEC.BAT file at the beginning of this Guided Activity, skip the following three steps.

20. To return your AUTOEXEC.BAT file to normal, type **DEL AUTOEXEC.BAT** and press <ENTER>.

21. Type **REN AUTOEXEC.OLD AUTOEXEC.BAT** and press <ENTER>.

22. Reboot your computer system.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`MODE LPT1:`	Resets the redirected output back to parallel port 1.
`MODE LPT1:80,6`	Causes output to a parallel printer to be printed 80 characters wide with 6 lines per vertical inch.
`MODE LPT1:132,8`	Causes output to a parallel printer to be printed 132 characters wide with 8 lines per vertical inch.
`MODE LPT1:=COM1:`	Redirects output from parallel port 1 to serial port 1.
`PRINT`	After creating a print queue, generates a status report of files in the queue and those being printed.
`PRINT/D:PRN/B:10240/Q:5`	Creates a print queue of 10,240 bytes (10K) that directs output to printer port 1 (PRN) and can hold a maximum of five files.
`PRINT/T`	Cancels the printing of all files in the print queue.

REVIEW QUESTIONS

1. When setting the characters per line (to be printed by your printer) with the MODE command, what values may be used?

2. When setting the number of lines per vertical inch (to be printed by your printer) with the MODE command, what values may be used?

3. What command form would be used to redirect output from LPT3: to COM3:?

4. After using the command form in question 3, how would you reset your computer system so that output to the printer was again directed to LPT3:?

5. What function does the PRINT command serve?

6. How can all files in a print queue be cancelled?

7. How can a single file (called LAST.DOC) be cancelled from a print queue?

8. If a standard device is not stated in the PRINT command, which device is assumed?

9. Can global wildcard characters be used in specifying files with the PRINT command?

10. What is the distinction between foreground and background computing tasks? Is the PRINT command queue a foreground or background task?

DOCUMENTATION RESEARCH

Check Chapter 7 of the *IBM Disk Operating System Version 3.30 Reference* manual or Chapter 3 of the *Microsoft MS-DOS User's Reference* to answer these questions.

1. What response is given by DOS if the /D, /B, or /Q parameter of the PRINT command is given a second time in a computing session?*

2. What happens if the PRINT command encounters a disk error when reading a file to be printed?*

3. What happens to the resident size of DOS the first time the PRINT command is executed?*

4. What is the maximum length in characters of a print queue entry including the drive name?**

* Not covered in the MS-DOS documentation. ** Not covered in the PC DOS documentation.

APPLICATION I

PART 1: RESETTING PRINTER OUTPUT WITH THE MODE COMMAND

In Unit 11, we discussed the use of the MODE command in printer control. In this application, you will create a file and then use the MODE command to print it out in two type sizes.

A. Create a small text file using EDLIN. It should be between 10 and 15 lines in length.

B. Use the MODE command to set the printer to 132 characters wide with 8 lines per vertical inch.

C. With the paper set to the top of the form, print your file to the printer. Do not advance the paper when you are finished printing.

D. Use the MODE command to set the printer to 80 characters wide with 6 lines per vertical inch.

E. Print your file. Both print jobs should appear on the same page.

F. Turn in the printed page to your instructor.

UNIT 12

RAM DRIVES AND DATA COMMUNICATIONS

OBJECTIVES

After completing this unit, you will be able to

1. explain the MODE command parameters associated with the communications protocol, including baud rate, parity, data bits, and stopbits;
2. define a virtual drive;
3. rate the speed differences among a floppy disk drive, a fixed disk drive, and a virtual drive.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. MODE
2. VDISK or RAMDRIVE.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): baud rate, bit, byte, data bits, parity, protocol, RAM drive, stopbits, terminal software program, virtual disk, virtual drive, and word size.
____Guided Activity: Use the RAM Drive.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

THE MODE COMMAND--SETTING THE SERIAL PROTOCOL

One function of the MODE command is to set or initialize the parameters used by the serial port. These parameters are referred to as **protocol** for data communications. The term protocol refers to a set of defined rules governing the exchange of data between two computer systems (micro to micro, micro to mainframe, or mainframe to micro).

Computers that communicate using data communications must speak the same language. They must speak at the same speed (**baud rate**), speak in uniform **word sizes** (**data bits**), and know when each word or packet of data ends (**stopbits**). The two computers must also know how error checking occurs so that signals scrambled over telecommunication lines, often telephone lines, may be repeated and corrected. **Parity** is a device used for error checking.

The baud rate indicates the number of changes in an analog signal per second. At or below 600 baud, the baud rate is roughly the same as the number of **bits** per second (bps) transmitted to the receiving computer. At higher speeds, the baud rate and the bps may differ. A more accurate measure of transmission speeds is bps, which represents the number of binary digits (1s or 0s) transmitted each second. Remember that serial communications transfers one bit at a time. Baud rates are established by the hardware (modem), and standardized rates exist. Rates supported by the MODE command are 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19,200. To designate the baud rate, you can use just the first two digits of the rate; thus, 12 means 1200 baud and 19 means 19,200 baud. Most PCs use modems operating at 300 or 1200 baud rates, though 2400 baud modems are becoming more popular as the cost drops.

There are three common parity check options. There may be no error checking, or parity may be set as either even or odd. Data is communicated in blocks. Each block includes bits of data represented by binary digits. Their value is either zero (0) or one (1). If even parity checking is used, the parity bit is set to either 0 or 1 in order to make the value of the entire transmitted block "even." If parity is odd, the reverse happens. The receiving computer then checks the parity of each block of

data received. If it does not match the expected parity, a message is sent to retransmit the last block of data.

Word length is set by the data bit's size; that is, either seven or eight bits. Those seven or eight bits represent a character. The receiving computer must know how long each "word" is. A computer not knowing word length can be compared to your trying to reading a book without spaces between the words. You would find it frustrating, confusing, and difficult to interpret.

The syntax of this form of the MODE command is

```
MODE COMn[:] b[,p][,d][,s][,P]
```

You must substitute values for the letters listed in the above syntax diagram.

`n` = serial port number
`b` = baud rate
`p` = parity
`d` = data bit value
`s` = stopbits
`P` = printer retry parameter

You must specify the baud rate, but other settings are optional. Remember, with DOS 3.x you can have up to four serial ports numbered COM1:, COM2:, COM3:, and COM4:. The baud rates include 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19,200. Your parity options include EVEN, NONE, and ODD. Word length or data bit size may be seven or eight. Stopbits are set to one or two. The retry parameter tells DOS that the settings are for a serial printer connected to the serial port. Further, it indicates that if a time-out error is encountered, the system should continuously retry the port until data can be received by the printer.

If the MODE command were used to set the baud rate as 1200, parity at EVEN, data bits at seven, stopbits at one, and with continuous retries if time-out errors occur, the following command form would be appropriate:

```
MODE COM1: 1200,E,7,1,P
```

The command could also be entered as

```
MODE COM1: 12,E,7,1,P
```

Most computer users involved in telecommunications use a **terminal software program**. This type of software makes it easier to set the communications protocol without using the MODE command. You may never have to do more than select options from a terminal program menu, and the software then invokes the MODE command for you. That is why this section is presented as background information and no exercise is provided.

THE DOS RAM DRIVE

Three types of disk drives are used with PCs. The two most obvious ones are floppy disk drives and hard disks (sometimes referred to as fixed disks). Both of these types must be formatted first, and subsequently used as secondary storage. Data may be stored including the application program code, the operating system code (DOS), and data (text files, data base files, etc.).

The third type of disk is less tangible and less easily understood. It is the **virtual disk** or **RAM drive**. PC DOS 3.x provides a file called VDISK.SYS (MS-DOS uses the filename RAMDRIVE.SYS), which can be called from the configuration file (CONFIG.SYS) to create a **virtual drive**. Once created, a virtual drive can be treated as any other disk drive or hard disk, with a couple of exceptions:

1. virtual drives never need to be formatted before use;
2. virtual drives are volatile memory or storage.

A virtual drive is created from a portion of memory. Files may be copied to it or from it. However, when you reboot the computer or if the system loses power, any data stored on the virtual drive is lost.

Why are virtual drives used? A floppy drive is a slow method of transferring data to and from the CPU. It must get "up to speed" before data can be read. That takes milliseconds, which ultimately slows down the process time. A hard disk is faster, but is still a mechanical device. It operates faster than the floppy drive, but slower than a virtual drive. There are no mechanical parts to a virtual drive. It does not have to get up to speed. It operates at the speed of electrons. Any electronic device is faster than a mechanical device. Speed is what makes the virtual drive attractive.

The virtual drive is created in conjunction with the DEVICE command and placed in the CONFIG.SYS file.

> **NOTE:** This command is not generated via the keyboard, but from the CONFIG.SYS file, which is read immediately after the resident portion of DOS is loaded and before the AUTOEXEC.BAT file is read.

To install the virtual drive, the following syntax is used

```
DEVICE=VDISK.SYS [k] [s] [d][/m][:st]
```

where k is the size of the virtual drive in kilobytes and s represents the number of **bytes** per sector on the virtual disk. The d parameter indicates the maximum number of directory listings contained on the virtual drive. One directory listing equates to one file. If you intend to have 12 files on the virtual drive, the directory parameter must be set to 12 or a larger number.

NOTE: MS-DOS uses the RAMDRIVE.SYS command rather than VDISK.SYS.

The default settings for the VDISK command are represented by issuing the command

```
DEVICE=VDISK.SYS 64 128 64
```

This would create a RAM disk 64K in size, with 128 bytes per sector, and capable of holding 64 directory entries (files).

NOTE: If using MS-DOS, the default values for RAMDRIVE.SYS are identical. The default settings for the RAMDRIVE command are represented by issuing the command

```
DEVICE=RAMDRIVE.SYS 64 128 64
```

The /m parameter tells DOS what type of memory to use in the creation of the virtual drive. To place the virtual drive in extended memory--that is, memory above the 1MB level--you would use the /E parameter. To use expanded memory (MS-DOS only)--that is, memory swapping using the Lotus-Intel-Microsoft specification--you would use the /A parameter. If you fail to stipulate expanded (MS-DOS only) or extended memory, DOS defaults to using conventional memory (bottom 640K) in your computer for the virtual drive. The :st parameter refers to the number of sectors transferred at one time from the virtual drive to the CPU. The :st parameter is only available with PC DOS and uses a default setting of 8. Other available choices in PC DOS include the values 1, 2, 3, 4, 5, 6, or 7.

If you are operating in extended memory with a virtual disk, the possibility exists that some interrupts may be lost. Interrupts are the way that hardware and software try to get the attention of the CPU. An interrupt says, "There's another task for the CPU to handle, so stop what you are doing." Sometimes the number of sectors transmitted at one time must be reduced to minimize the length of time the transfer takes because interrupts are being lost.

To create a virtual disk composed of 64 kilobytes, 128-byte sectors (transmitted four sectors at a time), and 32 directory entries in extended memory, the command form would be

```
DEVICE=VDISK.SYS 64 128 32/E:4
```

The MS-DOS equivalent (minus the sectors transmitted option) would be

```
DEVICE=RAMDRIVE.SYS 64 128 32/E
```

When the virtual drive is installed with DOS 3.x, the screen displays the following message:

```
VDISK Version 3.3 virtual disk x:
```

where x is the drive designation of the virtual drive. The first virtual drive is always given a drive designation one higher (in the alphabet) than the highest drive name. If your computer has two floppy disk drives and a hard disk, then DOS assigns the virtual drive the name D:. On the other hand, the virtual drive would become B: if your system only had a single floppy drive. If you had two hard disks (C: and D:), the first virtual drive becomes drive E:.

GUIDED ACTIVITY

USE THE RAM DRIVE: In this Guided Activity, you will create a small (64K) virtual drive and then copy files to the drive. You will also graphically see that any data stored on a virtual drive is lost when the computer system loses power or is rebooted. Drive D: is used for the virtual drive in this Guided Activity, which assumes that your personal computer has only one hard-disk drive.

> **CAUTION:** If you are working on a PC connected to a network, do not continue with this exercise unless given specific directions by your instructor. Depending on the network software and configuration, you may not be able to complete this Guided Activity.

> **NOTE:** If you are using MS-DOS, substitute RAMDRIVE.SYS every time you are instructed to enter VDISK.SYS.

1. Be sure you are logged on to the root directory of your hard disk.

2. The assumption is made that your AUTOEXEC.BAT file opened a path (using the PATH command) to the directory containing your DOS files.

The next few steps will protect your existing CONFIG.SYS file. You will return the file to its original condition by the time this exercise is completed.

> **NOTE:** You could simply edit the CONFIG.SYS file with EDLIN, and a backup file (CONFIG.BAK) would be automatically created. CONFIG.BAK would save the CONFIG.SYS file in its original form--unless you edited it again. For that reason, we are instructing you to take an extra precaution and make a backup copy of CONFIG.SYS under the filename CONFIG.OLD.

3. Type **`COPY CONFIG.SYS CONFIG.OLD`** and press <ENTER>.

4. Type **`DIR CONFIG.*`** and press <ENTER>.

You should see at least two files, CONFIG.SYS and CONFIG.OLD. There may be a third file called CONFIG.BAK.

In this exercise, you will need to reboot your computer and not have it execute the AUTOEXEC.BAT file. The next step renames it so it is not executed. You will return it to its original name by the end of the exercise.

5. Type **REN AUTOEXEC.BAT AUTOEXEC.OLD** and press <ENTER>.

6. Type **CLS** and press <ENTER>.

7. Type **EDLIN CONFIG.SYS** and press <ENTER>.

The screen will display the EDLIN prompt as follows:

```
End of input file
*
```

8. Type **L** and press <ENTER> to list the file.

9. Type **#I** and press <ENTER> to insert a new last line in the file.

10. Type **DEVICE=C:\DOS\VDISK.SYS 64 128 12** and press <ENTER>.

11. Press <^C>.

The monitor display should appear like the one shown in Fig. 12-1.

```
C:\>EDLIN CONFIG.SYS
End of input file
*L
       1:*BREAK=ON
       2: BUFFERS=20
       3: FILES=20
       4: DEVICE-C:\DOS\ANSI.SYS
*#I
       5:*DEVICE=C:\DOS\VDISK.SYS 64 128 12
       6:*^C

*
```

Fig. 12-1. Screen display when creating the CONFIG.SYS file.

12. Type **E** and press <ENTER>.

13. To reboot your system, simultaneously press the <CTRL>, <ALT>, and <DEL> Keys.

Assuming that you have a microcomputer system with one or two floppy disk drives and a single hard disk, the message displayed below will appear.

```
VDISK Version 3.30 virtual disk D:
Buffer size:          64 KB
Sector size:          128
Directory entries:    12
```

14. Enter the date and time.

15. Type **CLS** and press <ENTER>.

16. Type **DIR D:** and press <ENTER>.

Figure 12-2 shows the screen response you should receive.

```
C>DIR D:

 Volume in drive D is VDISK  V3.3
 Directory of  D:\

File not found

C>
```

Fig. 12-2. Empty directory listing on drive D:.

Next, you will copy two files created in Unit 9 to the virtual drive. The disk was formatted with the label AUTOFORMAT.

17. Place the data disk from the Unit 9 Guided Activity in drive A:.

18. Type **CLS** and press <ENTER>.

19. Type **COPY A:*.* D:** and press <ENTER>.

20. Type **D:** and press <ENTER>.

21. Type **DIR** and press <ENTER>.

The screen should resemble that shown in Fig. 12-3 and display at least two files now on your virtual drive.

```
C>COPY A:*.* D:
A:FILELIST
A:FINDTEXT
        2 File(s) copied

C>D:

D>DIR

 Volume in drive D is VDISK  V3.3
 Directory of  D:\

FILELIST         2275   5-28-89   5:11p
FINDTEXT          250   5-28-89   5:20p
        2 File(s)     61696 bytes free

D>
```

Fig. 12-3. Directory listing of virtual drive D:.

Notice that the number of bytes free (61,696) when added to the bytes used by the two files equals 64KB. That is the size of the drive you set up when using the VDISK command.

22. Type **CLS** and press <ENTER>.

At this point you could go in and edit one of the files using EDLIN. However, the PATH command must first be used to open a pathway between the default drive (D:) and the drive with the EDLIN external DOS command (C:).

23. Type **PATH C:\DOS** and press <ENTER>.

24. Type **EDLIN FINDTEXT** and press <ENTER>.

The screen will display the EDLIN prompt and the following message:

```
End of input file
*_
```

25. Type L and press <ENTER>.

Remember this text from an earlier exercise in Unit 9? The monitor display will appear as shown in Fig. 12-4. You could edit this file, but instead just quit and exit.

```
D>PATH C:\DOS

D>EDLIN FINDTEXT
End of input file
*L
       1:*This is a file in which to
       2: test the FIND Filter provided
       3: by DOS 3.0 and higher
       4:
       5: To test or not to test...
       6: that is not the question.
       7: Rather, the question is whether
       8: Find can find the word test
       9: or if the test will be failed.
      10:
      11: TEST PATTERN!
*
```

Fig. 12-4. Listing FINDTEXT file on virtual drive D:.

26. Type Q and press <ENTER>.

27. When prompted to "Abort edit (Y/N)?" type Y

28. Open the door to disk drive A:. You do not need to remove the disk; merely open the drive door.

In the next few steps, you will check to see if the virtual drive data is volatile!

29. To reboot your system, simultaneously press the <CTRL>, <ALT>, and <DEL> Keys.

30. After the power-on self-test concludes, the screen will indicate that the virtual drive is installed. Enter the time and date when prompted.

31. Type **DIR D:** and press <ENTER>.

The files are gone. They disappeared when the warm boot turned the system off.

The final steps will reset the AUTOEXEC.BAT and CONFIG.SYS files to their original condition.

32. Type **REN AUTOEXEC.OLD AUTOEXEC.BAT** and press <ENTER>.

33. Type **DEL CONFIG.SYS** and press <ENTER>.

34. Type **REN CONFIG.OLD CONFIG.SYS** and press <ENTER>.

35. Reboot your system.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`DEVICE=RAMDRIVE.SYS 64 128 64`	Creates an MS-DOS virtual drive when this command is added to the CONFIG.SYS file.
`DEVICE=VDISK.SYS 64 128 64`	Creates a PC DOS virtual drive when this command is added to the CONFIG.SYS file.
`MODE COM1:12,E,7,1`	Sets the serial port protocol to accept and transmit data at 1200 baud with even parity, seven data bits as the word length, and one stopbit.

REVIEW QUESTIONS

1. What are the five parameters used with the MODE command to set the communications protocol?

2. What is baud rate, and which rates are most common for PC users?

3. What is parity? What does it do?

4. Why is data bit size important to telecommunications?

5. What is a virtual drive? How is it created?

6. What is the advantage of a virtual drive?

7. What disadvantages exist for a virtual drive?

8. Which has the slowest access time--a hard disk, virtual drive, or floppy disk drive? Which has the fastest access time? Why?

DOCUMENTATION RESEARCH

Check Chapter 4 of the *IBM Disk Operating System Version 3.30 Reference* manual or pages 282-283 in the *Microsoft MS-DOS User's Reference* to answer these questions.

1. What is the minimum size (in kilobytes) of a virtual drive?*

2. What is the maximum size (in kilobytes) of a virtual drive?**

3. Should a RAM drive or virtual disk be formatted before use?**

4. What is the largest size RAM disk installable in extended memory?**

5. Is it possible to install more than one virtual drive?

* The answer is different for MS-DOS than for PC DOS.

** Not covered in the MS-DOS documentation.

APPLICATION J

PART 1: CREATING MULTIPLE VIRTUAL DRIVES

In Unit 12, you modified a CONFIG.SYS file to create a single virtual drive. In this application, you will modify the CONFIG.SYS file so that it creates two virtual drives.

A. Copy the CONFIG.SYS file from your hard disk to a formatted diskette.

B. Use EDLIN to modify the copy of CONFIG.SYS. Be sure to make the changes on your floppy disk--not on the hard-disk drive.

C. Modify the CONFIG.SYS file in order to create two virtual drives. The first virtual drive should use the DOS default values. The second virtual drive should be 6K in size, use 512-byte sectors, and have the capability of storing a maximum of 10 files.

D. Save the modified CONFIG.SYS file to disk. Turn in the disk to your instructor.

UNIT 13

THE DEBUG COMMAND

OBJECTIVES

After completing this unit, you will be able to

1. list the functions of the DEBUG command;
2. define the difference between the decimal and hexidecimal numbering systems;
3. view the contents of a disk sector or a segment of memory.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): binary, binary files, hexadecimal, and offset.
____Guided Activity: View Data with DEBUG.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

THE DEBUG COMMAND

DEBUG is to **binary files** what EDLIN is to line-oriented text files. It is an edito program provided with DOS. DEBUG can be used to view, create, modify, an execute .COM or .EXE files.

EDLIN, as you learned in Unit 6, is used to edit one line of text at a time--thus, it name EDLIN or Line Editor. On the other hand, DEBUG is used to modif individual bytes of data or program files.

A more complete list of the uses and functions of DEBUG would include th following:

1. to display the contents of a disk sector;
2. to display the contents of a segment of memory (RAM);
3. to display the contents of a file;
4. to edit files;
5. to recover lost or erased files;
6. to write and assemble program code.

NOTE: It is not within the purview of this text to teach programming. We will not teach DEBUG as a tool for writing programs. This unit is meant as a "quick and dirty" survey of DEBUG and will only cover the more common features used by non-programming DOS users. To that end, only the first four uses of the DEBUG command will be covered briefly.

WHY DO THEY TEACH THE HEXADECIMAL NUMBERING SYSTEM IN COLLEGE?

Any numbers used in conjunction with the DEBUG command must be based on th **hexadecimal** numbering system. DEBUG does not understand either **binary** o decimal numbers. Hexadecimal (usually shortened to hex) is a numbering systen using 16 as its base rather than 10 (which is used by the decimal numbering system) Hex is handy because programmers can represent the contents of any eight-bi binary number with only two digits. In other words, any printing or non-printabl character can be represented by a two-digit hex number. The computer really onl understands binary numbers (represented by the presence or absence of a magneti charge--0 or 1). The same single character could be represented by an eight-digi number in binary or a three-digit number in decimal notation. Thus, representin characters in hex is more efficient.

A simple comparison of the first 32 characters in the base 10 (decimal) and base 1 (hex) numbering systems may help you see the relationship between the two.

Decimal Value	Hexadecimal Value
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F
32	20

Notice that the decimal numbers repeat after nine (9) while the hex values do not begin repeating until after F. F is the highest single-digit value in hex (having a decimal value of 15).

The hexadecimal numbering system and its comparison to the binary and decimal systems are discussed for one reason. The use of the DEBUG command requires that decimal values or ASCII codes be converted into hexadecimal numbers. Figure 13-1 includes the ASCII (decimal) and hexadecimal values for most common characters. Remember that the decimal values for characters were used in Unit 10

to redefine keys using the ANSI.SYS driver and the PROMPT command. Yo already have some exposure to these values (or scan codes) in their decimal forma

Char	Hex	ASCII	Char	Hex	ASCII	Char	Hex	ASCII
!	21	33	@	40	64	a	61	97
"	22	34	A	41	65	b	62	98
#	23	35	B	42	66	c	63	99
$	24	36	C	43	67	d	64	100
%	25	37	D	44	68	e	65	101
&	26	38	E	45	69	f	66	102
'	27	39	F	46	70	g	67	103
(	28	40	G	47	71	h	68	104
)	29	41	H	48	72	i	69	105
*	2A	42	I	49	73	j	6A	106
+	2B	43	J	4A	74	k	6B	107
,	2C	44	K	4B	75	l	6C	108
-	2D	45	L	4C	76	m	6D	109
.	2E	46	M	4D	77	n	6E	110
/	2F	47	N	4E	78	o	6F	111
0	30	48	O	4F	79	p	70	112
1	31	49	P	50	80	q	71	113
2	32	50	Q	51	81	r	72	114
3	33	51	R	52	82	s	73	115
4	34	52	S	53	83	t	74	116
5	35	53	T	54	84	u	75	117
6	36	54	U	55	85	v	76	118
7	37	55	V	56	86	w	77	119
8	38	56	W	57	87	x	78	120
9	39	57	X	58	88	y	79	121
:	3A	58	Y	59	89	z	7A	122
;	3B	59	Z	5A	90	{	7B	123
<	3C	60	[	5B	91	|	7C	124
=	3D	61	\	5C	92	}	7D	125
>	3E	62	]	5D	93	~	7E	126
?	3F	63	^	5E	94			

Fig. 13-1. Hexadecimal and binary values corresponding to ASCII characters.

Additionally, you need to know the corresponding hex values for such things as line feed, form feed, and carriage return. Why? Because as you view or modi files using DEBUG, you need to know specifically which text character or devic control code is represented by each hex value. The following table lists som additional keystrokes, their ASCII decimal and hex values, and their effects.

Keystroke	Hex	ASCII	Effect
CTRL-@	00	00	Null
CTRL-G	07	07	Beep
CTRL-H	08	08	Backspace
CTRL-I	09	09	Tab
CTRL-J	0A	10	Line feed
CTRL-K	0B	11	Cursor to home position
CTRL-L	0C	12	Form feed
CTRL-M	0D	13	Carriage return
CTRL-Z	1A	26	End-of-file mark
CTRL-[	1B	27	Escape
ESC	1B	27	Escape
CTRL-\	1C	28	Right arrow
CTRL-]	1D	29	Left arrow
CTRL-^	1E	30	Up arrow
CTRL-_	1F	31	Down arrow
SPACEBAR	20	32	Space

Fig. 13-2. Hex and decimal values for device control codes.

Now that you have a chart to refer to, we can proceed with starting DEBUG and viewing a segment of memory or a disk sector.

INVOKING AND QUITTING DEBUG

There are two methods of invoking or starting DEBUG. The first is to simply enter DEBUG at the DOS prompt, followed by pressing <ENTER>. Once DEBUG is executed, it has its own prompt--just like EDLIN does. Remember that the EDLIN prompt was a lone asterisk (*). The DEBUG prompt is a dash (-). Once the DEBUG prompt appears, you can execute any of the DEBUG subcommands that will be discussed shortly.

The second method of invoking DEBUG follows this syntax:

```
DEBUG [d:][path][filename.ext]
```

If you wish to view a file named LOOKATIT.DOC, you would enter the command form

```
DEBUG LOOKATIT.DOC
```

If the desired file is not on the default drive, be sure to enter the drive and path. If LOOKATIT.DOC is on drive A: in the \WEST directory, and the default path is C:\DOS, you would enter the following command line:

```
DEBUG A:\WEST\LOOKATIT.DOC
```

If the specified file is not found, DEBUG returns the message

```
File not found
```

> **NOTE:** Even if the file is not found, you will end up at the DEBUG prompt with the DEBUG program active.

Either method loads the DEBUG command, but only the second method also loads a file into memory. Once loaded, LOOKATIT.DOC can be viewed, modified, and saved to disk.

Either of these methods will leave you at the cryptic DEBUG prompt displayed in Fig. 13-3.

```
C:\MAIL>DEBUG LOOKATIT.DOC
-
```

Fig. 13-3. The DEBUG command prompt.

To view or modify a file, you must know some additional DEBUG subcommands. However, before covering the additional commands, you will need to know how to exit DEBUG. DEBUG, like EDLIN, uses the Q or QUIT subcommand. To exit, simply type **Q** (in upper- or lowercase) and press <ENTER>. The result is shown in Fig. 13-4.

```
C:\MAIL>DEBUG LOOKATIT.DOC
-Q

C:\MAIL>
```

Fig. 13-4. The result of executing the DEBUG QUIT (Q) command.

ADDITIONAL DEBUG COMMANDS

All DEBUG commands are composed of a single letter. Some can accept additional parameters--some cannot. The following list provides a brief explanation of each command and its proper syntax. After the listing, the final sections of this unit will detail the more common DEBUG subcommands.

Keep in mind that all DEBUG commands may be entered in either upper- or lowercase. These subcommands are not case-sensitive.

A [address]	Assemble
C [range]	Compare
D [range]	Dump or display
E address[list]	Enter
F range list	Fill
G [=address [address...]]	Go
H value value	Hex
I value	Input
L [address][drive:record record]	Load
M range address	Move
N filename [filename]	Name
O value byte	Output
Q Quit	DEBUG
R [register]	Register
S range list	Search
T [=address][value]	Trace
U [range]	Unassemble
W [address][drive:record record]	Write

THE DEBUG DUMP SUBCOMMAND

The most basic DEBUG command uses the DUMP (D) subcommand. If you load a file when invoking the DEBUG command, you can use the DUMP subcommand to display the contents of the first 128 bytes of the file. Each time you press D, the next 128 bytes of the file will be displayed.

In Fig. 13-5, you see the results of loading DEBUG and a file named TECHNO.TXT (from the West Student Data Disk). After the file is loaded, the DUMP subcommand is invoked, causing eight lines of data to be displayed. Each line is divided into three components. The left portion of the screen includes a 64K segment address in memory (where these characters are stored--after the file is loaded) followed by a colon and an **offset**. These two numbers are in hex notation.

```
C:\>DEBUG A:TECHNO.TXT
-D
32EE:0100  54 65 63 68 6E 6F 6C 6F-67 79 20 69 73 20 61 20   Technology is a
32EE:0110  74 65 72 6D 20 63 75 72-72 65 6E 74 6C 79 20 61   term currently a
32EE:0120  73 73 6F 63 69 61 74 65-64 20 77 69 74 68 20 74   ssociated with t
32EE:0130  68 65 0D 0A 63 6F 6D 70-75 74 65 72 20 69 6E 66   he..computer inf
32EE:0140  6F 72 6D 61 74 69 6F 6E-20 61 67 65 2C 20 79 65   ormation age, ye
32EE:0150  74 20 65 61 63 68 20 67-65 6E 65 72 61 74 69 6F   t each generatio
32EE:0160  6E 20 68 61 73 0D 0A 64-65 76 65 6C 6F 70 65 64   n has..developed
32EE:0170  20 69 74 73 20 22 6E 65-77 22 20 74 65 63 68 6E    its "new" techn
-
```

Fig. 13-5. The results of executing the DEBUG DISPLAY (D) command.

The 64K segment address is relative. Depending on what else is in memory (RAM), the 64K segment address can vary each time you load the file. The offset indicates how far from the beginning of the file the current characters are found.

The middle portion of the screen includes the hex values for 16 characters in the file. These values are then translated in the right portion of the screen into their ASCII characters. If no ASCII printing character equivalent exists, the byte is represented by a period.

Each of the eight lines represents sixteen characters for a total of 128 bytes or 128 characters per segment. Each time you press D, DEBUG displays another 128-byte segment of the file. In essence, you can page through the entire file 128 bytes at a time.

When invoking the DUMP subcommand a second time, you would see a display similar to the one in Fig. 13-6.

```
C:\>DEBUG A:TECHNO.TXT
-D
32EE:0100  54 65 63 68 6E 6F 6C 6F-67 79 20 69 73 20 61 20   Technology is a
32EE:0110  74 65 72 6D 20 63 75 72-72 65 6E 74 6C 79 20 61   term currently a
32EE:0120  73 73 6F 63 69 61 74 65-64 20 77 69 74 68 20 74   ssociated with t
32EE:0130  68 65 0D 0A 63 6F 6D 70-75 74 65 72 20 69 6E 66   he..computer inf
32EE:0140  6F 72 6D 61 74 69 6F 6E-20 61 67 65 2C 20 79 65   ormation age, ye
32EE:0150  74 20 65 61 63 68 20 67-65 6E 65 72 61 74 69 6F   t each generatio
32EE:0160  6E 20 68 61 73 0D 0A 64-65 76 65 6C 6F 70 65 64   n has..developed
32EE:0170  20 69 74 73 20 22 6E 65-77 22 20 74 65 63 68 6E    its "new" techn
-D
32EE:0180  6F 6C 6F 67 79 2E 20 20-50 72 6F 64 75 63 74 73   ology.  Products
32EE:0190  20 74 68 61 74 20 77 65-0D 0A 61 73 73 75 6D 65    that we..assume
32EE:01A0  20 68 61 76 65 20 61 6C-77 61 79 73 20 62 65 65    have always bee
32EE:01B0  6E 20 61 76 61 69 6C 61-62 6C 65 2C 20 6D 61 79   n available, may
32EE:01C0  20 68 61 76 65 20 62 65-65 6E 0D 0A 69 6E 76 65    have been..inve
32EE:01D0  6E 74 65 64 20 6A 75 73-74 20 61 20 64 65 63 61   nted just a deca
32EE:01E0  64 65 20 6F 72 20 74 77-6F 20 61 67 6F 20 61 6E   de or two ago an
32EE:01F0  64 20 77 65 72 65 0D 0A-63 6F 6E 73 69 64 65 72   d were..consider
-
```

Fig. 13-6. The second 128-byte DUMP subcommand display.

If you continue pressing D, you will page through all segments of memory containing the TECHNO.TXT file and then continue to view memory that contains other programs (memory resident) or the residue of previously used data or program code.

> **NOTE:** Keep in mind that once a file is loaded with DEBUG, if the D subcommand is invoked with no parameters, the first 128 bytes are displayed beginning at offset 100. If you do not enter a filename when invoking DEBUG, the D subcommand displays the first 128 bytes of the first available memory segment beginning at offset 100.

You can use a range of addresses or an offset with the DUMP subcommand. It has several optional formats that can be used.

You can specify the offet at which you want data displayed. To display the next 128 bytes beginning at offset 2000, you would enter

```
D 2000
```

You can also instruct DEBUG to display the first 128 bytes beginning at any address in RAM. For instance, the following command displays the first 128 bytes beginning at segment address 32EE:

```
D 32EE:0000
```

Another form of the command allows you to view data between two offset values. For instance, the following command displays the data between offset 0160 and 01D0:

```
D 0160 01D0
```

The final option enables you to display a given amount of memory beginning at a specified offset of the current segment. To view 48 bytes (hex 30) beginning at offset 1000, you would enter

```
D 1000 L 30
```

GUIDED ACTIVITY

VIEW DATA WITH DEBUG: In this Guided Activity, you will examine the TECHNO.TXT file found on the West Student Data Disk using the DEBUG command.

1. Boot your system, and go to the directory containing your DOS files, probably C:\DOS.
2. Insert your West Student Data Disk in drive A:.
3. Type **CLS** and press <ENTER>.
4. Type **DEBUG A:TECHNO.TXT** and press <ENTER>.
5. Type **D** and press <ENTER>.

Your screen should look similar to the one in Fig. 13-7, except that the address on the far left of each line will probably be different. Your offset should match the offset shown in the figure.

```
C:\DOS>DEBUG A:TECHNO.TXT
-D
32EE:0100  54 65 63 68 6E 6F 6C 6F-67 79 20 69 73 20 61 20   Technology is a
32EE:0110  74 65 72 6D 20 63 75 72-72 65 6E 74 6C 79 20 61   term currently a
32EE:0120  73 73 6F 63 69 61 74 65-64 20 77 69 74 68 20 74   ssociated with t
32EE:0130  68 65 0D 0A 63 6F 6D 70-75 74 65 72 20 69 6E 66   he..computer inf
32EE:0140  6F 72 6D 61 74 69 6F 6E-20 61 67 65 2C 20 79 65   ormation age, ye
32EE:0150  74 20 65 61 63 68 20 67-65 6E 65 72 61 74 69 6F   t each generatio
32EE:0160  6E 20 68 61 73 0D 0A 64-65 76 65 6C 6F 70 65 64   n has..developed
32EE:0170  20 69 74 73 20 22 6E 65-77 22 20 74 65 63 68 6E    its "new" techn
-
```

Fig. 13-7. Display of first 128 bytes of TECHNO.TXT file.

6. Type **D** and press <ENTER>.

Figure 13-8 shows the screen display after invoking the DUMP subcommand a second time. Notice that the text of the file can be read in the far right portion of the screen.

```
C:\DOS>DEBUG A:TECHNO.TXT
-D
32EE:0100  54 65 63 68 6E 6F 6C 6F-67 79 20 69 73 20 61 20   Technology is a
32EE:0110  74 65 72 6D 20 63 75 72-72 65 6E 74 6C 79 20 61   term currently a
32EE:0120  73 73 6F 63 69 61 74 65-64 20 77 69 74 68 20 74   ssociated with t
32EE:0130  68 65 0D 0A 63 6F 6D 70-75 74 65 72 20 69 6E 66   he..computer inf
32EE:0140  6F 72 6D 61 74 69 6F 6E-20 61 67 65 2C 20 79 65   ormation age, ye
32EE:0150  74 20 65 61 63 68 20 67-65 6E 65 72 61 74 69 6F   t each generatio
32EE:0160  6E 20 68 61 73 0D 0A 64-65 76 65 6C 6F 70 65 64   n has..developed
32EE:0170  20 69 74 73 20 22 6E 65-77 22 20 74 65 63 68 6E    its "new" techn
-D
32EE:0180  6F 6C 6F 67 79 2E 20 20-50 72 6F 64 75 63 74 73   ology.  Products
32EE:0190  20 74 68 61 74 20 77 65-0D 0A 61 73 73 75 6D 65    that we..assume
32EE:01A0  20 68 61 76 65 20 61 6C-77 61 79 73 20 62 65 65    have always bee
32EE:01B0  6E 20 61 76 61 69 6C 61-62 6C 65 2C 20 6D 61 79   n available, may
32EE:01C0  20 68 61 76 65 20 62 65-65 6E 0D 0A 69 6E 76 65    have been..inve
32EE:01D0  6E 74 65 64 20 6A 75 73-74 20 61 20 64 65 63 61   nted just a deca
32EE:01E0  64 65 20 6F 72 20 74 77-6F 20 61 67 6F 20 61 6E   de or two ago an
32EE:01F0  64 20 77 65 72 65 0D 0A-63 6F 6E 73 69 64 65 72   d were..consider
-
```

Fig. 13-8. Display of the second 128 bytes of TECHNO.TXT file.

7. Type **D** and press <ENTER> until you come to the end of the file. It will probably take about 14 entries of the D (DUMP) subcommand to reach the end of the file. The end of the file will be indicated by the hex value 1A.

Figure 13-9 shows the end of the file, as indicated by hex value 1A, at the end of the 4th line in the final 128-byte segment.

```
32EE:0820  6B 61 6E 20 6F 69 6C 0D-0A 73 70 69 6C 6C 2C 20   kan oil..spill,
32EE:0830  62 69 6F 2D 64 65 67 72-61 64 61 62 6C 65 20 70   bio-degradable p
32EE:0840  6C 61 73 74 69 63 20 63-6F 6E 74 61 69 6E 65 72   lastic container
32EE:0850  73 2C 20 74 65 73 74 2D-74 75 62 65 0D 0A 62 61   s, test-tube..ba
32EE:0860  62 69 65 73 2C 20 73 75-72 72 6F 67 61 74 65 20   bies, surrogate
32EE:0870  6D 6F 74 68 65 72 68 6F-6F 64 2C 20 67 65 6E 65   motherhood, gene
-D
32EE:0880  74 69 63 20 65 6E 67 69-6E 65 65 72 69 6E 67 2C   tic engineering,
32EE:0890  0D 0A 61 6E 64 20 74 68-65 20 38 30 34 38 36 20   ..and the 80486
32EE:08A0  6D 69 63 72 6F 70 72 6F-63 65 73 73 6F 72 2E 20   microprocessor.
32EE:08B0  20 54 68 65 20 6C 61 74-74 65 72 20 65 6E 61 62    The latter enab
32EE:08C0  6C 65 73 0D 0A 70 65 72-73 6F 6E 61 6C 20 63 6F   les..personal co
32EE:08D0  6D 70 75 74 65 72 20 6D-61 6E 75 66 61 63 74 75   mputer manufactu
32EE:08E0  72 65 72 73 20 74 6F 20-62 75 69 6C 64 20 61 0D   rers to build a.
32EE:08F0  0A 6D 61 69 6E 66 72 61-6D 65 20 63 6F 6D 70 75   .mainframe compu
-D
32EE:0900  74 65 72 20 74 68 61 74-20 77 69 6C 6C 20 66 69   ter that will fi
32EE:0910  74 20 6F 6E 20 61 20 64-65 73 6B 74 6F 70 2E 20   t on a desktop.
32EE:0920  20 0D 0A 0D 0A 57 68 61-74 20 77 69 6C 6C 20 74    ....What will t
32EE:0930  68 65 20 31 39 39 30 73-20 62 72 69 6E 67 3F 1A   he 1990s bring?.
32EE:0940  20 44 61 74 61 20 52 65-63 6F 72 64 73 0D 0A 20    Data Records..
32EE:0950  20 20 20 20 20 20 20 34-2E 36 20 20 20 20 20 20          4.6
32EE:0960  20 20 20 20 4D 75 6C 74-69 70 6C 65 20 46 69 6C      Multiple Fil
32EE:0970  65 73 0D 0A 20 20 20 20-20 20 20 20 34 2E 37 20   es..        4.7
-
```

Fig. 13-9. The Hex value 1A marks the end of the file.

8. Type **D** and press <ENTER> one more time. Notice that there is data in memory beyond the end of TECHNO.TXT. Usually the data will include a large number of spaces represented by hex 20. It may be garbage, or it may be the residue of a previously loaded file. It all depends on the commands executed prior to using DEBUG in this exercise.

9. Type **D 700** and press <ENTER> to dump the file beginning at offset 700. The data should look like those shown in Fig. 13-10.

```
32EE:0920  20 0D 0A 0D 0A 57 68 61-74 20 77 69 6C 6C 20 74    ....What will t
32EE:0930  68 65 20 31 39 39 30 73-20 62 72 69 6E 67 3F 1A   he 1990s bring?.
32EE:0940  20 44 61 74 61 20 52 65-63 6F 72 64 73 0D 0A 20    Data Records..
32EE:0950  20 20 20 20 20 20 20 34-2E 36 20 20 20 20 20 20          4.6
32EE:0960  20 20 20 20 4D 75 6C 74-69 70 6C 65 20 46 69 6C      Multiple Fil
32EE:0970  65 73 0D 0A 20 20 20 20-20 20 20 20 34 2E 37 20   es..        4.7
-D
32EE:0980  20 20 20 20 20 20 20 20-20 51 75 69 74 74 69 6E           Quittin
32EE:0990  67 20 54 68 69 73 20 50-72 6F 67 72 61 6D 0D 0A   g This Program..
32EE:09A0  20 20 20 20 0D 0A 35 2E-30 20 20 20 20 20 20 20      ..5.0
32EE:09B0  20 20 20 20 20 20 55 73-61 67 65 20 48 69 6E 74        Usage Hint
32EE:09C0  73 0D 0A 20 20 20 20 20-20 20 20 35 2E 31 20 20   s..         5.1
32EE:09D0  20 20 20 20 20 20 20 20-50 72 69 6E 74 69 6E 67           Printing
32EE:09E0  20 47 72 6F 75 70 73 20-6F 66 20 45 6E 76 65 6C    Groups of Envel
32EE:09F0  6F 70 65 73 0D 0A 20 20-20 20 20 20 20 20 35 2E   opes..         5.
-D 700
32EE:0700  6D 70 75 74 65 72 2C 0D-0A 6D 69 63 72 6F 77 61   mputer,..microwa
32EE:0710  76 65 73 2C 20 66 6F 6F-64 20 70 72 6F 63 65 73   ves, food proces
32EE:0720  73 6F 72 73 2C 20 63 61-62 6C 65 20 74 65 6C 65   sors, cable tele
32EE:0730  76 69 73 69 6F 6E 2C 0D-0A 6D 75 73 69 63 20 76   vision,..music v
32EE:0740  69 64 65 6F 73 2C 20 61-6E 64 20 74 68 65 20 66   ideos, and the f
32EE:0750  69 72 73 74 20 73 70 61-63 65 20 73 68 75 74 74   irst space shutt
32EE:0760  6C 65 20 69 6E 63 72 65-61 73 65 64 0D 0A 6F 75   le increased..ou
32EE:0770  72 20 65 78 70 65 63 74-61 74 69 6F 6E 73 20 6F   r expectations o
-
```

Fig. 13-10. Data at offset 700.

10. Type `Q` and press <ENTER> to quit DEBUG.

11. Type `CLS` and press <ENTER>.

ANOTHER METHOD OF LOADING A FILE WITH DEBUG

There may be times when you wish to invoke DEBUG and display data in memory before loading a specific file. Or you may have forgotten to load the file with DEBUG. In either case, the NAME (N) and LOAD (L) subcommands are used to load a specific file after DEBUG is already active. The process includes assigning a filename to match the filename on disk (with the NAME subcommand) and then invoking the LOAD subcommand. An example follows.

```
DEBUG
N A:\WEST\LOOKITUP.DOC
L
```

In the first line DEBUG is invoked. The second line uses the N command to set the filename. Finally, the third line loads the named file using the L subcommand.

The above three lines perform the same function as

```
DEBUG A:\WEST\LOOKITUP.DOC
```

MODIFYING DATA WITH DEBUG

DEBUG can be used to modify files, in addition to viewing their contents. Two additional subcommands are needed to change data and code using DEBUG. The first is the ENTER (E) subcommand. Its syntax is

```
E address datastring
```

The address can be the offset if you are modifying the default segment of memory. The datastring can be entered in either hex notation or ASCII characters--if the data string is enclosed in quotation marks.

For instance, the following two entries are equivalent

```
E 150 "Change it!" 0D 0A
E 150 43 68 61 6E 67 65 20 69 74 21 0D 0A
```

Both versions will insert the text "Change it!" followed by a carriage return (0D) and a line feed (0A) at offset 150.

You can also use the ENTER subcommand followed only by the offset to display the current contents of a single byte and change it or bypass it. The form of the command would be

```
E 150
```

Pressing the <SPACEBAR> leaves the byte of data unchanged, or you can enter the hex notation for a new character.

The second subcommand that allows you to modify text or code is the FILL (F) subcommand. It is used to initialize a segment of memory so you can enter or write the data you desire.

The FILL subcommand uses the following syntax:

```
F range fillerdata
```

The range is an offset range, and the fillerdata is the characters to be written into the range of bytes indicated by the offset range. For instance, you could enter spaces in the bytes between offset 100 and 900 by using the command form

```
F 100 900 20
```

Once you have written zeroes or spaces to a range of bytes, you are free to write new data to that range of memory with the ENTER subcommand. By initializing the memory segment, you avoid the residue characters that may cause you problems if incorporated into the data or code you are writing.

PROGRAMMING AND MODIFYING DATA AND CODE WITH DEBUG

DEBUG is a powerful tool for the programmer. It can be used

1. to write assembly language programs or routines;
2. to recover lost files still in memory;
3. to recover deleted or lost files on disk;
4. to edit existing data files or program files.

Most of these functions are beyond the scope of this book, but a brief exposure to the remaining DEBUG subcommands can be informative.

DEBUG can be used to assemble and unassemble machine-language programs. The commands providing these functions are ASSEMBLE (A) and UNASSEMBLE (U), respectively.

Once you assemble program code, it can be executed or run using the GO (G) subcommand.

Several block operations are supported by DEBUG. The COMPARE (C) subcommand can be used to compare one segment of memory with another. The MOVE (M) subcommand can transfer a range of data in memory to a specific beginning address.

DEBUG also provides a SEARCH (S) subcommand to locate a given list of bytes. The list is composed of one or more bytes of data in hex notation or ASCII (if surrounded by quotations). When hex notation is used, the notation used for each byte is delimited by spaces or commas. The SEARCH subcommand can locate a given string of characters in a data file, program, or register or in memory.

For instance, the following command would list the location of all occurences of a carriage return and line feed (0D 0A) within the current segment (offset 200 to 900).

```
S 200:900 L 0D 0A
```

DEBUG can also display the contents of one or more CPU registers. The 8088 CPU has 14 registers. We'll just mention three. The AX register is the accumulator and is used for arithmetic functions or operations. The CS register points to the code segment that is currently being executed. The DS register points to the data segment or area of memory where current data is stored.

To view the current contents of the CS register, you would enter the DEBUG subcommand

```
RCS
```

The R is the REGISTER command, and CS identifies which register's contents are to be displayed.

The syntax of the subcommand is

```
R[register]
```

ADDITIONAL DEBUG TOOLS

The DEBUG command includes several additional subcommands that are useful tools for programmers. The *Microsoft MS-DOS User's Reference* includes additional data on these subcommands in Chapter 8. However, the *IBM Disk Operating System Version 3.30 Reference* manual does not include any information on DEBUG.

REVIEW QUESTIONS

1. Describe the two ways of invoking DEBUG.

2. Which character is the DEBUG prompt?

3. List the command form that would load DEBUG while logged on to drive C: and allow you to load the file A:\TEST\THISISIT.TXT.

4. How could you view the first 128 bytes of a file loaded with DEBUG?

5. How could you view the first 128 bytes of a file beginning at offset 1000 (using the DEBUG command)?

UNIT 14

DOS LOCAL AREA NETWORK COMMANDS

OBJECTIVES

After completing this unit, you will be able to

1. define a local area network (LAN);
2. discuss the causes behind the development of LANs;
3. discuss the advantages of a LAN;
4. define some issues related to selection of a LAN;
5. define and explain the purposes of DOS 3.x commands that support networks;
6. discuss generally three types of Novell NetWare commands and understand how they are used.

IMPORTANT COMMANDS

The important commands introduced in this unit are

1. FCBS
2. LASTDRIVE
3. SHARE.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): cabling, command line utilities, console commands, disk server, file control block, file server, file-sharing, footprint, gateway, local area network (LAN), media, menu utilities, network administrator, node, private volume, public volume, superstation, topology, workstation, and workstation prompt.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

LOCAL AREA NETWORKS--THE WAVE OF THE FUTURE?

1986 was heralded as the year of the **local area network** (**LAN**). Though the adoption of a uniform standard for networks failed to materialize, manufacturers of LAN-related hardware and software made great strides forward. With the appearance of PC DOS 3.x, IBM and Microsoft provided support via the operating system for such a system.

Before describing the commands in DOS 3.x that support networking, a working definition of a LAN and a discussion of the market forces that created a need for LANs are appropriate.

In the simplest form, a LAN is a system for connecting multiple workstations (i.e., microcomputers). A LAN facilitates the sharing of data, files, and peripherals. Originally LANs were contrasted with wide area networks, which used telecommunications to connect workstations. The systems are not as distinct today; a LAN may include hard-wired or cable linkages as well as linkages via modem.

Why did the LAN evolve? PCs gave new-found freedom to end users. With the greater variety of application software, tasks that had previously been handled by the data processing department were now performed at each worker's desk. But even before the euphoria faded, it became clear that data on an isolated workstation could not be transported conveniently to another workstation.

Workers at different workstations often need to use the same data base or the same spreadsheets. To share data, files had to be copied to disk and transported to another workstation before they could be used. Soon it became unclear which machine had the most recent, modified version of the file. Time was lost determining which version was accurate. Ultimately, there had to be a better way of sharing files. The LAN served that purpose. Shared files were stored either on a **file server** or a **disk server**. A file server is a hard disk reserved specifically for shared files on the network. It is a separate and distinct resource on the network. A disk server is a hard disk in one user's PC which is linked into the system for

shared access by anyone on the network. A disk server is less expensive to implement than a dedicated file server, but access speed is slower.

When a disk server is used, you may encounter the terms **workstation** and **superstation**. A superstation is the PC that contains the disk server and is used by the network administrator to maintain the network system. It is also often used as a regular workstation for normal computing operations.

A contributing factor to the development of LANs was the high cost of peripherals. If a department had three workstations, each needing access to a printer, and the work required only a dot-matrix printer, the cost was not terribly restrictive. But if each workstation needed access to true letter-quality reproduction, then the cost of peripherals skyrocketed. Further, if both graphics and letter-quality type were needed, then each workstation required two printers, a dot-matrix and a letter-quality. Another alternative is the laser printer, but even with recent price drops, the cost is in the $1700 to $3500 range. A related factor is the additional **footprint** (i.e., work space) required to handle a computer, two printers, and other necessary peripherals.

The LAN solved this problem by permitting users to access any printer or peripheral networked into the system. Thus, a single laser printer or a single letter-quality printer could meet the needs of a department or work group.

A complete discussion of LANs cannot be covered in a single chapter. The installation of a network is not a simple matter. A few topics related to network selection include

1. **Topology**: How are the workstations to be connected? The standard topologies include token ring (adopted by IBM), star, and bus.

2. **Cabling**: What type of cable or **media** are to be used? This relates to the speed at which data can be communicated across the LAN. Types include twisted pair, baseband coaxial, broadband coaxial, and fiber optics.

3. Security: Can the system be password-protected? Can each volume or disk volume be declared public or private?

4. Software: Are there network versions of the application software available for use on the system?

5. Peripherals: What hardware can be shared via the network? Who should have access to the equipment? Does the system permit the restriction of certain peripherals to designated users?

Other issues might include the following: Will an electronic mail system be included? Do the network linkages require additional hardware and software, or just software and the use of an existing serial port? Who will serve as **network administrator** for the LAN; that is, who will assign passwords, monitor usage, and

provide technical support? Is there a limitation on the distance between **nodes** (i.e., workstations) in the system? Will the system include a **gateway** (that is, access to the enterprise's mainframe system or another network)?

NETWORK SUPPORT FIRST OFFERED BY PC DOS 3.X

DOS 3.1 provided disk, directory, and printer support for most DOS commands on several LANs, including IBM PC-Net and Microsoft's MS-NET. With DOS 3.2, specific support features for IBM's token ring network were added. Each version of DOS 3.x provides three commands specifically supporting networks: SHARE, FCBS, and LASTDRIVE.

THE SHARE COMMAND

The SHARE command performs a straightforward function. It loads support for **file-sharing** and file change protection. Three factors are controlled: the number of files that may be open at one time, the number of locks, and the allocation of space for recording data necessary for file-sharing. When the SHARE command is activated, it checks the **file control block** (FCB) table. The FCB table is created by using the FCBS command within the CONFIG.SYS file. Remember that CONFIG.SYS is always referenced when the computer system is booted.

The simplest form of the command is

```
SHARE
```

Since SHARE is an external command, the user must be logged on to the drive and directory that include the SHARE.EXE command file, or must specify the drive and path prior to the command. The following example assumes that the SHARE.EXE file is on drive C: in the DOS subdirectory but the default drive is not C:. This form also uses two default settings for the command.

```
C:\DOS\SHARE
```

If the default settings are not appropriate, then parameters must follow the command. The two parameters or switches are

/F:bytesize	This switch sets the file size for a special file that stores data necessary for file-sharing. For each file to be opened, the special file must be 11 bytes long plus the length of the full filename. The default is 2048 bytes.
/L:locks	This switch allocates space for the number of locks desired. The default size is 20 locks. Locks are devices used by DOS to limit access to records. If more than one user has read/write access to the same record at the same time, data corruption may result.

If we spelled out the default settings in the SHARE command, it would be entered via the keyboard as

```
C:\DOS\SHARE/F:2048/L:20
```

The SHARE command is invoked only once each time the system is booted. If it is invoked a second time, DOS will issue a message stating "SHARE already installed".

You will find that some network software includes its own version of the SHARE command--such as the 10NET system.

THE FILE CONTROL BLOCK COMMAND

The second network-related command is the FCBS (File Control BlockS) command. It is always used within the CONFIG.SYS file and follows the form

```
FCBS=m,p
```

where m equals the maximum number of files that may be open at one time and p equals the protected number of files. FCBS is used only when file-sharing is activated using the SHARE command.

If no FCBS command is entered, DOS assumes the following default values:

```
FCBS=4,0
```

which indicates that a maximum of four files may be open and that none of them is protected. However, if the SHARE command is invoked, and you have not included the FCBS command in the CONFIG.SYS file, DOS assumes the default value for FCBS as shown below.

```
FCBS=16,8
```

This new default means, that if an application program opens 16 files during the course of its processing, the limit is not exceeded and all processing continues. However, if subsequently the application needs to open one or more files, a dilemma ensues. DOS has been told not to open more than 16 files, but the application is greedy for number 17. DOS then surveys the files in use to determine the least recently used file and proceeds to boot that file out of memory to make room for the new file, with one exception. Because of the FCBS command, the first 8 files loaded into memory are exempt from getting the boot, even if they would otherwise be the least recently used files. A few guidelines are in order:

1. In the FCBS command, the maximum number of files open (m) should never be exceeded by the number of protected (p) files.

2. If the maximum number of files (m) equals the number of protected files (p), then an error message will appear stating "FCB unavailable Abort, Fail", and processing will cease.
3. If two or more FCBS commands refer to the same file on a network, then DOS counts the multiple FCBS commands as only one.
4. If an application experiences critical errors because of FCB files being closed, the maximum number of files (m) that may be open at one time should be increased.
5. Whenever the FCBS command is included in the configuration file (CONFIG.SYS), the resident size of DOS is increased.

SPECIFYING THE LASTDRIVE

The final DOS network-related command is LASTDRIVE. It, like FCBS, is executed via the CONFIG.SYS file and is used to expand the number of drives that may be accessed by the system. Without specifying LASTDRIVE, a system may access drives A: through E:.

Throughout this text, we have referred to drives A:, B:, and C:. These three drive designations usually refer to two floppy disk drives and a hard disk. We also discussed the addition of a virtual disk. A system with two floppy drives and a hard disk would designate the virtual disk as drive D:. The naming of additional drives continues up the alphabetic ladder. Some systems may include all of the above plus a tape backup unit. That unit might be referred to as drive E:.

Unless specified otherwise, the PC system running under DOS 3.x assumes the last drive name available is E:. In other words, the LASTDRIVE command is automatically set to equal E unless you indicate otherwise. Microcomputers connected to a network may find this a limitation, so DOS provides the option of increasing the LASTDRIVE value.

It is not uncommon for networked PCs to have both public (shared) and private (limited access) drives or directories (also referred to as **public volumes** and **private volumes**). In fact, some networks divide a single hard disk into multiple directories, each designated as a different drive for a different user. Some of the volumes (or partitions) may be public (accessible to anyone on the network), while others may be private (accessible only from the local workstation).

In addition to drives being declared public or private, drives are also considered either local or network (non-local) in nature. If your workstation has two floppy drives and a hard disk, all three are considered local drives. That is, they are physically part of your workstation. If you need to access a hard disk that is physically connected to another workstation on the network, that hard disk is considered a network or non-local drive.

NOTE: Each local or network drive is declared during network configuration as either a public or a private drive. When we discuss using PC DOS and MS-DOS commands on a network, an important consideration is whether you are trying to use the commands on a local or a network drive. Some DOS commands will work on a local drive, but not a network drive.

Let us examine two stations on a multistation network.

STATION ONE	STATION TWO
Floppy drive = A:	Floppy drive = A:
Floppy drive = B:	Floppy drive = B:
Hard drive (public) = C:	Hard disk (private) = C:
RAM disk = D:	RAM disk = D:
Tape backup unit = E:	

The user on station two has access to six drives. When working on station two, the following drive designators could be used:

STATION TWO SETUP

Drive Designator	Status	Actual Equipment
A:	local	Floppy drive A: on Station Two
B:	local	Floppy drive B: on Station Two
C:	local	Hard disk (private) on Station Two
D:	local	RAM disk on Station Two
E:	network	Hard disk (public) on Station One
F:	network	Tape backup unit on Station One

The drive designations are specific to each workstation on the network. Drive D: on station one is not the same as drive D: on any other workstation on the network. Below you will see that station one does not share its drives with any other workstation on the network, but does share its tape backup unit.

STATION ONE SETUP

Drive Designator	Status	Actual Equipment
A:	local	Floppy drive A: on Station One
B:	local	Floppy drive B: on Station One
C:	local	Hard disk (public) on Station One
D:	local	RAM disk on Station One
E:	local	Tape backup unit on Station One

Further, station two has drives designated beyond the default value of E:. In order for this configuration to work, the LASTDRIVE command must exist in the CONFIG.SYS file. Its form would be

```
LASTDRIVE=F
```

The syntax of the LASTDRIVE command is

```
LASTDRIVE=x
```

where x is equal to the last drive name. The value must be alphabetic, and the greatest value available would be Z.

There does not appear to be any gain or loss in memory or functionality by raising the LASTDRIVE value higher than necessary. For instance, no damage results from making LASTDRIVE equal to Z, even if only the traditional A:, B:, and C: drives are physically present.

A typical CONFIG.SYS file for a workstation on a network might include the following commands:

```
FILES=50
BUFFERS=15
LASTDRIVE=H
```

At such a station, the user would have access to eight logical drives (A through H).

NETWORK SOFTWARE COMMANDS

The software that is specific to each commercial network includes many additional commands. Network software typically includes modified versions of the DOS commands discussed throughout this text, in addition to its own library of commands. After you know DOS, you at least have a head start in learning network software commands.

Though it is impossible in one unit to cover all the network commands (often called utilities), an examination of one network software system might be instructional. In the remainder of this unit, we will discuss the Novell NetWare System, describing how DOS commands function on a network and examining select NetWare commands.

WHAT IS NOVELL NETWARE?

Each network includes two components: hardware and software. Hardware includes network cards or some type of interface card, the connecting cable, and a disk or file server. Common network interface cards include Ethernet, ArcNet, and Token Ring.

The software overlays DOS and provides utility commands that enables users to create, modify, and access data and applications on the network. Common network software includes Novell, 3COM, PC-NET, MS-NET, and 10NET.

In essence, the network software provides an additional dimension to the operating system that creates an environment where multiple users may access data, applications, and peripherals in an orderly fashion. On any multiuser system it is important for each user to maintain appropriate rights while not infringing upon the rights of other users to create, store, modify, and protect data and to access system resources (memory and peripherals).

Novell's NetWare is a typical but powerful network software package. It can be used with a variety of network or interface cards.

USING DOS COMMANDS ON A NOVELL NETWARE SYSTEM

Novell's NetWare software is designed to operate in an MS-DOS/PC DOS environment. NetWare runs under versions of DOS 2.x, 3.x, and 4.x. Of almost 40 primary commands under DOS 3.x, over half work equally well on either a local drive or a network drive, assuming the network workstation is PC compatible. The remaining commands work only on a local drive.

When considering the use of DOS commands on network drives, keep in mind that NetWare can include both PC compatibles and Macintosh computers on a single network.

For the remainder of this section, we will consider DOS commands that operate differently (or not at all) on a network drive. This survey is presented as an example of how DOS can function differently on a network as opposed to a stand-alone PC.

Each of the following commands works normally on a Novell network unless you include a volume or file server name in the directory path. If you need to include a volume or file server name in the directory path, use the NetWare command listed in the right column below.

COMP	
COPY	NCOPY
DIR	NDIR
DEL	
ERASE	
MKDIR or MD	
RMDIR or RD	

The next group of commands works the same on a Novell network (both network and local drives) unless the network drive is on a Macintosh workstation--in which case a specific NetWare command exists to work with the Macintosh network drive.

BACKUP This command works effectively with PC-compatible network drives, but not Macintosh network drives. To back up data using the Macintosh file structure, use the NetWare MACBACK command.

RENAME The RENAME command works only if you use standard DOS naming conventions; that is, up to eight-character filenames with up to three-character extensions. For longer filenames, as used by the Macintosh, you must rename the files from a Macintosh workstation.

XCOPY The Macintosh resource fork is not copied by the DOS XCOPY command. To copy Macintosh files, use the NetWare NCOPY command.

The following commands only work correctly on local drives. You may not use them on network drives.

ASSIGN
DISKCOMP
DISKCOPY
FORMAT
SUBST

This last group of commands comprises a special category of DOS commands. They are troublesome on a Novell network, are used only in limited situations, or have a more effective NetWare counterpart.

CHKDSK	NetWare provides the CHKVOL command for use on network drives. It functions in the same way as CHKDSK. The DOS CHKDSK command should be used only on local drives.
LABEL	The DOS LABEL command will attempt to delete all files in the root directory. This command should never be used on a Novell network. If you accidentally use it, try to recover your files with the NetWare SALVAGE command.
PATH	This command works correctly on both local and network drives. However, the NetWare MAP command may be more helpful and convenient to use.
PRINT	The DOS PRINT command may be used only to send local or network files to a local printer from the attached workstation.
RECOVER	The RECOVER command can be used only on local drives. For network drives you must use the NetWare SALVAGE command.
RESTORE	If you wish to restore files to the same directory from which they were backed up, you can use the DOS RESTORE command. Otherwise, you must use LARCHIVE and LRESTORE (for local drives) and NARCHIVE and NRESTORE (for network drives).

Other network software often appears to use DOS commands equally well on either a local or a network drive. However, they may actually install new versions of the DOS commands that have been modified to work in the network environment. This is also true of some multiuser system software.

TYPES OF NOVELL NETWARE COMMANDS

In the next section, we will discuss three types of specific NetWare commands. All of the commands are considered **command line utilities**. They may be accessed by users at the **workstation prompt**. The workstation prompt (also called the command line) is the equivalent of a DOS or system prompt on a stand-alone PC. Novell also includes **menu utilities** that accomplish the same functions, but provide a series of screens or menu options to lead you through a task.

Menu utilities are usually used by less experienced users because of the "hand holding" they provide. Experienced users choose speed and can accomplish the same tasks faster with command line utilities. As an example, the FILER menu utility or the NDIR command may be used to view files in a specific directory.

Finally, Novell's NetWare includes a series of **console commands** that are used by the network administrator to monitor and control various file server functions. These functions include providing file server security, setting the file server clock, sending messages to all logged-in users, establishing or changing print queue priorities, and regulating network printers and other network resources.

SELECTED NOVELL NETWARE COMMANDS

The following brief coverage is provided to give you a flavor of what specific Novell NetWare commands exist. The list is not complete, but offers an overview of the types of functions needed to make a network functional.

ATTACH Users on networks may have files and applications stored on a local drive or a network drive. A network drive may be a public volume on another workstation's hard disk or a volume on a file server. A file server is basically a dedicated hard disk for the storage of public and private data. Networks may include multiple file servers. The ATTACH command is used to attach your workstation to a file server. Once attached, you can access the data and applications stored there if the volume is public or if you are authorized to access the data. Authorization is typically via password or sign-on code.

CAPTURE Most non-network application software sends output directly to a local (attached) printer. A workstation on a network often does not include a local printer. The CAPTURE command is used to capture output intended for a local printer and redirects the output to a network (non-local) printer. In essence it tricks the application program into thinking there is an attached local printer. The CAPTURE command has numerous parameters. Among its features are the ability to direct output to a given file server (with an attached network printer), the capability of forcing a form feed at the end of each document or print session, and the ability to convert tabs to eight character spaces. CAPTURE also enables you to specify a banner (first printed page) that identifies the printed document as originating from your workstation.

CASTOFF Networks enable users of one workstation to send messages to other workstations. If your workstation receives a message, it cannot execute additional actions until the message is cleared (by pressing <CTRL-ESC>). CASTOFF prevents messages from reaching your console. This command is used when you wish to begin executing either an unattended process or a long process. For instance, CASTOFF might be invoked before compiling a program, reorganizing keys, or using data communications to establish a remote hookup.

CASTON CASTON enables your workstation console to receive messages from other stations on the network.

FLAG This command allows an authorized network user to set up to eight attributes for specific files. Four of these attributes were discussed earlier in connection with the DOS ATTRIB command. They include

Read-only or Read-write enabled
System
Hidden
Modified (the archive bit used by the BACKUP command)

Additionally, the Novell NetWare FLAG command can establish a file as executable only (cannot be modified), shareable or non-shareable (the latter means only one user may access the file at a time), indexed, or transactional. Large files (larger than 2MB) may be marked as indexed in order to increase access speed. Actually the bytes that compose the file are indexed in the file allocation table. Transactional files are those that require the network operating system to keep track of all changes and modifications to the file.

GRANT This NetWare command grants trustee rights to users or groups of NetWare users; that is, staff with similar levels of clearance or functionally similar access needs. The GRANT command offers eight types of trustee rights:

the right to read data files
the right to write data files
the right to open data files
the right to create files and subdirectories
the right to delete files and subdirectories
the right to modify the trustee rights
the right to search and display lists of files
the right to modify directory or file attributes and rename files or subdirectories

A user or class of users may be granted some, all, or none of these rights.

HOLDON This command holds the file(s) you are accessing open and denies write access to other network users. Other users may still have read-only access to the file(s). This feature is turned off by issuing the HOLDOFF command.

LOGIN The LOGIN command is used to gain access to a specific file server. The command invokes a login script for a specific file server and gains access to that server's resources (data, applications, and peripherals). Your rights at login are limited by your effective rights on that file server. That is, if you have been limited to read-only rights on a specific file server, you may not modify data on that server.

When you boot your workstation on the network, you are automatically and transparently attached to a specific file server. That file server is your default server. The LOGIN command may be used to access resources on another file server. Of course, your login script could automatically attach your station to multiple servers initially. Access to additional servers is controlled by password.

You may also detach your workstation from a specific file server by using the LOGOUT command.

MAP

For convenience, it is easier to refer to a volume, subdirectory, or path on a network using just a drive designation. For instance, you may wish to access the main file server and files in the SYS volume. To access data in the root directory of the SYS volume, you would have to precede each filename with

```
MAIN/SYS:
```

In all likelihood, you may wish to read data in other subdirectories below. In such a case, the file GOODNEWS.DOC would be addressed as

```
MAIN/SYS:SMART/DATA/GOODNEWS.DOC
```

The MAP command allows you to assign a drive designation to the path so that you are not required to type a long path string.

With MAP, an entire path may be equated with a drive designator. For instance, the following command greatly reduces the keystrokes needed to access GOODNEWS.DOC.

```
MAP M:=MAIN/SYS:SMART/DATA
```

After invoking this command form, you could access GOODNEWS.DOC by entering

```
M:GOODNEWS.DOC
```

PSTAT

Before printing a document, it is helpful to know if the target network printer is on-line or off, if it is busy, or what type of form it has mounted. The PSTAT command provides you with a status report on the printer.

PURGE Files may be marked for deletion by using the DOS DEL or ERASE commands. Further, the Novell FILER menu utility allows users to erase files. However, the use of the PURGE command makes those files irretrievable. Many DOS utilities such as Norton Utilities, Mace Utilities, and PC Tools allow you to undelete files (in some cases) that have been accidentally deleted. Further, a file server has a limited number of file and directory entries. The DEL and ERASE commands do not free up that link between the directory listing and the file--but the PURGE command does. PURGE only purges files deleted from your workstation.

RENDIR This command enables you to change or rename subdirectories.

As you can tell, the additional complexities of a network serving multiple users require additional utilities. Once a PC system moves beyond the stage of serving a single user, the rights of all users must be considered. Sharing computer resources, like sharing anything in life, necessitates rules and guidelines so that no one user suffers at the hands of another--either intentionally or accidentally.

COMMAND REVIEW DICTIONARY

In this unit, you have read about several DOS commands. Most have optional parameters or switches. We have tried to cover both the common and the rarely used forms of each command. The following forms of this unit's commands are the most frequently used and should be reviewed carefully:

`FCBS=16,8` Used within the CONFIG.SYS file. This form of the command establishes a maximum of 16 files that may be open at one time. The command further indicates that the first eight files loaded into memory have priority and will not be closed automatically in order to open a new file.

`LASTDRIVE=F` Indicates the last drive name that can be legally accessed by DOS. Without using the LASTNAME command, the default is drive E:; that is, only drives A:, B:, C:, D:, and E: may be accessed.

`SHARE` Loads support for file-sharing. This form uses the default settings.

REVIEW QUESTIONS

1. What is the impetus behind the growth in local area networks?
2. Define local area network.
3. Why is file-sharing important?
4. List at least five topics to be considered before selecting or implementing a local area network.
5. What function does a network administrator perform?
6. What support for networks was added with DOS 3.x?
7. What function does the SHARE command serve?
8. What function does the FCBS command serve?
9. What function does the LASTDRIVE command serve?

DOCUMENTATION RESEARCH

In the *IBM Disk Operating System Version 3.30 Reference* manual or the *Microsoft MS-DOS User's Reference*, read about the FCBS and LASTDRIVE commands, and check the index before answering the following questions.

1. What is the maximum number of open files that may be specified in the FCBS command?
2. What is the minimum number of open files that may be specified in the FCBS command?
3. What is the range of values (minimum and maximum number of files specified by FCBS) that may be opened and cannot be automatically closed by the operating system?*
4. What purpose do file control blocks serve?
5. Is the FCBS command effective when the computer system is not operating in a file-sharing (network) environment?*
6. Is the FCBS command executed from the keyboard? If so, when? If not, where is it used?
7. What happens if you issue the following FCBS command?*

```
FCBS=10,10
```

8. What happens if LASTDRIVE is set to less than the number of physical drives installed on your computer?*

* Not covered in the MS-DOS documentation.

UNIT 15

MULTITASKING, MULTIPROCESSING, AND THE FUTURE OF DOS

OBJECTIVES

After completing this unit, you will be able to

1. define the five user concerns for the future of DOS;
2. define multitasking;
3. define multiprocessing;
4. define upward compatibility;
5. explain the difference between expanded and extended memory.

ASSIGNMENTS

The assignments to be completed for this unit are

____Review and understand the following computer terms listed in the Computer Terms Dictionary (Appendix A): bank switching, central processing unit or CPU, conventional memory, DOS compatibility box, emulation, expanded memory, extended memory, multiprocessing, multitasking, Presentation Manager, upward compatibility, user-friendly, and user interface.
____Review Questions.
____Documentation Research.

Place a check in front of those assigned by your instructor.

FIVE USER CONCERNS FOR FUTURE DOS DIRECTIONS

As users are transformed from novices into experienced users, it is natural that they begin considering not only what DOS can do but also what it should be able to do. Five areas for enhancement are often mentioned. Most discussion of future DOS features centers on the following:

1. ability to use more memory for applications and data
2. multitasking capabilities
3. multiprocessing capabilities
4. upward compatibility
5. user-friendliness.

AVAILABILITY OF MORE MEMORY

The first area of concern can be illustrated by the following anecdote. A manager has an IBM PC on his desk. The machine is linked to a Burroughs A10 mainframe computer using an InterCom 1000+ **emulation** board produced by InterComputer Communication (ICC) Corporation. The card uses DataComm software to emulate the Burroughs; that is, it makes the desktop computer look and act like a mainframe terminal. The same computer has a Juniper II card, produced by Rohm Corporation, which gives the computer access to a modem pool for telecommunications. The Juniper II software stays resident in memory at all times. This manager uses the SmartWare II software produced by Informix Software Inc. Further, several memory-resident programs are used, including a public domain screen-saver utility. The manager goes to spell-check a document and sees this screen message: "Insufficient memory to load spellchecker".

Because of the way the Intel 8088 microprocessor addresses memory, microcomputers using that chip are limited to 1MB (1024K) of potential memory. In designing DOS, the developers earmarked 640K of the 1MB for application programs and their data. The remaining 384K are used for various tasks such as ROM-BIOS, video memory, and the POST (see Fig. 15-1). The 640K accessible to user programs and data are called user memory or **conventional memory**.

When you purchase a 640K computer system, the memory listed is user memory--not total system memory. The full megabyte is broken into 16 different 64K memory blocks. Figure 15-1 illustrates the different blocks of memory and their functions.

Memory Block Number	Memory Range	Uses of Memory
0	0-64K	Includes DOS and BASIC work area (when loaded), and keyboard buffer. With DOS 3.x and earlier, DOS consumed in the range of 24 to 40K.
1	64-128K	User work area.
2	128-192K	User work area.
3	192-256K	User work area.
4	256-320K	User work area.
5	320-384K	User work area.
6	384-448K	User work area.
7	448-512K	User work area.
8	512-576K	User work area.
9	576-640K	User work area.
10	640-704K	Extended memory for video display.
11	704-768K	Standard memory for video display.
12	768-832K	XT hard disk and EGA board ROM expansion.*
13	832-896K	Cartridge memory on PCjr.
14	896-960K	Cartridge memory on PCjr.
15	960-1024K	System ROM-BIOS, POST, and ROM-BASIC area.

* This memory typically resides on the hard-disk controller card or the enhanced graphics adapter (EGA) board.

Fig. 15-1. Sixteen blocks of PC memory and related functions.

In the anecdote about the manager, all available memory was in use. Conventional memory was consumed by loaded portions of DOS (including internal DOS commands), application programs, data, and memory-resident programs. No memory space remained into which the spellchecker program could be loaded.

Topping the list for most experienced or power users is the need for DOS to access more memory. As application programs expand capabilities, they take more memory. As the **user interface** becomes more friendly and graphic, it takes more memory. As users add more memory-resident functions (address books, auto-dialers, notepads, calendars, electronic mail, spelling checkers, thesauruses, DOS shells, and clock software), more memory is consumed.

Extended and expanded memory are two different methods of accessing larger amounts of memory than the PC was originally designed to use. Both types of non-conventional memory require software specifically written to take advantage of them.

EXTENDED MEMORY

While 8088/8086-based PCs are limited to 1MB of memory, the Intel 80286 chip can access 16MB of memory in protected mode. The 80286 can function in two modes. In real mode it acts as an 8088 chip running DOS programs--but considerably faster. In real mode, the 80286 can only access 1MB, just like the 8088 chip. However, in protected mode the 80286 microprocessor can access up to 16MB of memory. Unfortunately, in protected mode the chip cannot run programs created to run under DOS. To utilize protected mode, programs can be written to temporarily configure the CPU for extended memory, perform a task, and then reconfigure the CPU for DOS to use again.

The newer 80386 microprocessor also can access more than the 1MB of memory common to the 8088. The 80386 chip has three modes. It operates in real mode--performing just like a super-fast 8088 CPU. In protected mode, the 80386 provides the same support as an 80286 chip. It can address up to 16MB of memory but cannot run DOS applications, unless they are rewritten for protected mode. The third mode of the 80386 is native mode. It is a new mode allowing the CPU to access up to 4 gigabytes (GB) or 1024MB of memory. Programs written for the 8088 or 80286 cannot run under the 80386's native mode. The real advantage of the new native mode is that future application programs can access a tremendous amount of RAM--once the programs are written.

The ability to address extended memory (extended beyond the 1MB limitation) is built into the 80286 and 80386 microprocessors.

EXPANDED MEMORY

Expanded memory is the hope of the older 8088 and 8086 systems. It is an adaptation of a scheme known as **bank switching**. Banks of memory that are external to the CPU's 1MB memory space are swapped in and out of the memory area that DOS can access. Memory expansion boards supporting expanded memory are required to implement expanded memory.

The industry standard for implementation of expanded memory is the Lotus-Intel-Microsoft expanded memory specification (LIM EMS). 8088- and 8086-based systems can implement only EMS, not extended memory. However, 80286 and 80386 systems can implement either EMS or extended memory.

Either type of memory is useful only when used with software designed to effectively use the additional memory and follow its specifications.

Neither PC DOS 3.3 nor MS-DOS 3.3 supports LIM EMS. That support is not available until DOS 4.x.

MULTITASKING

Most experienced computer users still remember the initial glow of computing, when accomplishing anything on the machine was exciting. After the initial euphoria, the PC steadily becomes a valuable tool, but still just a tool. When that stage is reached, the manager, secretary, or clerk seeks to accomplish tasks better and faster rather than simply computerizing for the sake of computerizing. Once the tool is integral to the flow of business, the major concern becomes completing tasks faster. Time spent waiting for the computer to complete a task is unproductive time. Waiting for a large spreadsheet to recalculate or waiting for the printer to complete a long document is a bore, unless you can scurry around doing other non-computing tasks.

Multitasking is the capability of a microprocessor to handle several tasks at one time. Some software solutions have been created, such as internal print buffers, but the PC is still basically a single-task machine. The 8088 machines may be pretty much at the mercy of whatever software solutions DOS provides in this area, but the newer 80286 machines have the hardware (the microprocessor) to tackle multitasking. Multitasking is clearly the domain of OS/2. Except for the semblance of multitasking provided by the PRINT command, the 8088 and 8086 machines are single-task machines.

MULTIPROCESSING

Concurrent with the need to do more tasks at the same time, many businesses are discovering the need to handle multiple users at one time on a single machine. Future versions of DOS will tackle this problem by allowing the concurrent execution of multiple DOS work sessions on a single Intel 80386 CPU. In fact, in 1988 Intel announced great strides on its new 80486 chip. The 80486 is basically a mainframe computer on a single microchip. The ability to handle concurrent processing or multiprocessing will be a boon to both networked machines and multiuser configurations.

Non-IBM software and hardware are already available to create a multiuser machine out of a standard IBM PC or XT. Special interface cards are placed in the PC that link it to additional slave stations. With non-IBM vendor software, the system can support one user per slave station, with each user sharing the CPU, memory, disk drives, and other peripherals. A small department that needs only two or three workstations may find this type of configuration far more cost-effective than buying multiple PCs and networking them together. There are drawbacks to this type of multiuser system as it is currently available on PCs:

1. Often runs slower than non-multiuser systems.
2. Typically only two or three additional workstations may be linked to one PC. Beyond that, the system must be networked.
3. Uses a special system software that may conflict with other software in use.

4. Application software on the system may react unpredictably unless it is a network version or unless the multiuser software allows all but one user to be locked out of a file record or document.
5. Depending on the future direction of DOS, this equipment may become obsolete, unless the manufacturer is committed to continual enhancements and upgrades.

Ultimately, the solution for multiprocessing rests in the hands of the operating system developers. The microcomputer is no longer just a one-person machine. This is the one area not yet addressed by IBM's new OS/2. The expectation is that future versions of the operating system must add to and enhance the networking and multiuser capabilities of the PC.

UPWARD COMPATIBILITY

Individuals and businesses do not like to make a habit of purchasing expensive equipment that will shortly be obsolete. IBM pioneered the concept of families of computers, allowing users to upgrade to bigger, more powerful machines. Coupled with that concept is the assurance that new developments, in both software and hardware, will provide a migration path for users. No one should wake up to find his or her machine unsupported and unloved. Providing the capability to upgrade machines and avoid planned obsolescence is the essence of **upward compatibility**.

In a sense, OS/2 provides software upward compatibility. Software written for earlier versions of DOS can run on an OS/2 machine in what is referred to as a **DOS compatibility box**. Unfortunately, if OS/2 runs an old DOS application, that is all it can do. Use of the DOS compatibility box negates the operating system's ability to concurrently run other applications.

USER-FRIENDLINESS

Operating systems, almost by definition, have traditionally not been creatures for the novice user to grasp. Even students of the operating system find strange conventions, unusual syntax, and unfriendly user interfaces. PC DOS and MS-DOS have no help screens, and they sometimes use syntax that can befuddle an English major.

The intermediate solution is to acquire a **DOS shell** to make DOS less cryptic and more friendly. With the advent of OS/2 and DOS 4.x, the trend toward the use of graphic user interfaces, such as those used by Apple's Macintosh, has greatly accelerated. DOS 4.x (which is beyond the scope of this text) allows you to point and click with a mouse (pointing device), while OS/2 provides windows and icons rather than text-based screen choices. Future versions of DOS will certainly increase the use of **user-friendly** interfaces.

The development of the **Presentation Manager** and similar user interfaces has always been linked to overcoming several hardware barriers. Those barriers have

included increasing the amount of memory that can be addressed and developing a higher resolution graphic display and adapter.

CONCLUSION

In the long run, DOS must change and improve. New hardware and software advances dictate that the operating system must be growing and vital. Users who once were enamored of 640K (after starting with 64K machines) now think in terms of megabytes of RAM. Managers who once were satisfied with cold, unfriendly system prompts now want graphic, friendly user interfaces. Users are never satisfied, and shouldn't be, with yesterday's breakthroughs and enhancements.

ADDITIONAL REFERENCES ON DEVELOPMENTS IN PC DOS AND MS-DOS

The history of PC DOS and MS-DOS is written daily. It could be a full-time career keeping pace with new developments, enhancements, and bugs. The following publications are helpful in providing periodic updates on DOS:

INFOWORLD, P.O. Box 1018, Southeastern, PA 19398, published weekly.
PC MAGAZINE, P.O. Box 2443, Boulder, CO 80321, published biweekly.
PC WEEK, One Park Avenue, 4th Floor, New York, NY 10016, published weekly.
PC WORLD, P.O. Box 6700, Bergenfield, NJ 07621, published monthly.
PERSONAL COMPUTING, P.O. Box 2941, Boulder, CO 80321, published monthly.

REVIEW QUESTIONS

1. What is the difference between multitasking and multiprocessing?
2. How do expanded, extended, and conventional memory differ?
3. What is the driving force behind the need for multitasking?
4. What are the disadvantages of the current non-IBM multiuser configurations?
5. Define upward compatibility. Who developed the concept? Why is it important to hardware and software purchasers?

DOCUMENTATION RESEARCH

Because the *IBM Disk Operating System Version 3.30 Reference* manual and the *Microsoft MS-DOS User's Guide and User's Reference* do not address the issue of enhancements to future versions of DOS, documentation research for this chapter requires access to trade publications such as PC WEEK, PC WORLD, INFOWORLD, PC MAGAZINE, and PERSONAL COMPUTING.

1. Research and expand on one of the five areas of user concern for future versions of DOS.
2. Research and discuss the difference between Intel's 8088, 8086, 80286, and 80386 microprocessor chips. How do the differences affect the end user?
3. Research the new Intel 80486 chip.
4. What is OS/2? How is it different from DOS?

APPENDIX A

COMPUTER TERMS DICTIONARY

The terms contained in this dictionary are used throughout the text. Be sure to reference this section if a term is not defined within the text.

Allocation unit — A unit of storage on disk (see **Cluster**).

Alternative processing strategies — One facet of a disaster recovery plan that deals with finding other sites or machines for data processing in an emergency situation.

Archive — To back up or make a duplicate copy.

ATTRIB — The DOS command that flags a file as read-only or read-and-write. Also can flag a file as modified since the last backup.

Attribute bits — Each of the eight bits in the attribute byte. Depending on the setting (on or off) of each bit, the file will be write-protected, not write-protected, a system file, a hidden file, a subdirectory, a volume label, or set to be archived the next time you back up files.

Attribute byte — The eleventh byte of a directory listing which includes the read-only bit set by ATTRIB.

Background task

In the DOS environment, the task with lower priority is referred to as the background task. When competition arises between the foreground and background tasks over access to the CPU, the higher-priority task wins. Foreground tasks always win over background tasks.

Backup disks

Diskettes on which files created by the BACKUP command are placed.

Bank switching

A scheme used by expanded memory boards. In this strategy, the microprocessor cannot address more than 1MB (640K of usable RAM) at a time. The expanded memory is treated in a fashion similar to storing data on a disk. The most frequently accessed memory is stored in the lower 640K.

BASIC

Beginner's All-purpose Symbolic Instruction Code. A computer language developed at Dartmouth College to be used by students studying computer science.

Basic input/output system

BIOS is the portion of the disk operating system that is hardware dependent. It provides an interface between the operating system and the specific hardware (i.e. microprocessor) and peripherals (i.e., hard disk, disk drives, monitor, etc.) used.

Baud rate

A unit used to measure data communication speeds; that is, the rate at which data is communicated to a peripheral device or to another computer. Transmission speeds are listed in terms of baud rate or bits per seconds. Common rates for telecommunication include 300, 1200, and 2400 baud. However, DOS 4.x also supports 110, 150, 600, 4800, 9600, and 19,200 baud rates. Mainframes often communicate with their terminals at rates of 9600 baud or greater.

Bit

The smallest unit of storage or memory, represented by either the absence (0) or the presence (1) of electrical current. Bit is shorthand for binary digit. Eight bits equal one byte, and one byte typically represents one character, such as a number or letter.

Boilerplate documents — Standard blocks of text that are reused in numerous documents with only slight modification. A time-saving strategy.

Boot — The process of turning on a computer, which in turn loads the operating system and other software into RAM.

Bootable — A disk containing DOS system files and COMMAND.COM is said to be bootable. Those files are required to start up your system.

Boot record — Side 0, track 0, sector 0 of a disk. The boot record indicates which version of DOS is in use and contains a short program telling the computer to load the DOS system files.

Buffer — Part of the computer's RAM allocated for input and output operations. The buffer holds data until the peripheral device (to which it is directed) can accept the data.

Byte — A unit of storage (memory) composed of eight bits. One byte can represent one character, such as a letter or number.

Cabling — Refers to the type of connections used between workstations on a network. Types of cabling include twisted pair, coaxial, etc.

Case-sensitive — The condition in which uppercase and lowercase characters may not be freely substituted for each other.

Central processing unit — See CPU.

Clones — Personal computers that are manufactured to be compatible with an industry standard machine (like the IBM PC).

Cluster — A unit of storage on a fixed disk or diskette. A cluster may be composed of one or more sectors depending on the operating system and the type of media (hard disk, high-density disk, 360K floppy, etc.).

Code page

A table consisting of 256 different characters. Allows the IBM PC and PS/2 to provide national language support.

Command line utilities

Network commands executed at the workstation prompt.

Command prompt

See **DOS prompt.**

Compatibles

Often referred to as IBM clones or compatibles. These microcomputers are produced by non-IBM manufacturers who strive to make their machines run software exactly as a true IBM computer would. Compatibles often improve upon the basic IBM machine--e.g., faster clock speed, additional slots--while still striving for software compatibility.

Complete filename

Includes the drive designator, file specification, and file extension.

CON:

DOS reserved word for the console. The console is composed of both the monitor and the keyboard.

CONFIG.SYS

Configuration file used with DOS; must be contained in the root directory and may include several DOS commands, among them the FCBS, DEVICE, BUFFERS, and FILES commands.

Console commands

Network commands or utilities executed at the file server console. These commands are used by the network administrator to monitor and control various file server functions including providing security, setting the file server date and time, sending messages to logged-in users, establishing or changing print queue priorities, and regulating all network resources (memory, printers, peripherals).

Control characters

Special characters in a document that tell the printer to form feed, line feed, set margins, indent, or apply specific text attributes (such as boldface, italics, point size, font, etc.).

Copy-protection

A strategy by which the ability to duplicate software is restricted.

CPU — The central processing unit (i.e., the 8088, 80286, or 80386 microprocessor) or part of a computer that manipulates data and text.

Creation date — The date when a file was last modified. The accuracy of the creation date is contingent on the system date's being accurate when the file was last saved.

Creation time — The time when a file was last modified. The accuracy of the creation time is contingent on the system time's being accurate when the file was last saved.

Data bits — Another term for word size. Indicates if a character transmitted is seven or eight bits long.

Default drive — The drive designated by the DOS prompt. Unless indicated otherwise, DOS commands (such as CHKDSK, DEL, etc.) are invoked on the default drive.

Delimiter — A character used to indicate the beginning or end of a text string or parameter. Also used to separate the end of the file specification from the beginning of the filename extension.

Directory — A partition created by the user to store files. Directories and subdirectories are created to help organize files on a hard disk.

Directory table — The table stored on disk that stores such information as the file specification, extension, creation date and time, size in bytes, and attributes of a file. The directory table also stores a pointer indicating the beginning allocation unit where the file is stored on disk.

Disaster recovery plan — A comprehensive plan to anticipate an emergency that will interfere with normal computing and to propose alternatives. The basic component of every plan is periodic, complete system backups.

Disk server — A hard disk on one workstation in a network that is used by all workstations for storing files.

Display adapter

A card or board in the microcomputer that contains the circuitry to drive a monitor. Display adapters come in several varieties, including monochrome, color, graphics, and enhanced graphics. A monochrome adapter is required to drive a monochrome monitor. Likewise, a color display adapter is required to run a color monitor.

DOS

The disk operating system (DOS) is composed of coded instructions that tell the computer how to behave like a computer. These instructions cover management of internal communications, error checking, memory usage, data storage, data retrieval, data deletion, and device configuration.

DOS command

A word or phrase that executes a machine language routine stored in ROM, on disk, or in memory.

DOS compatibility box

A feature of OS/2 that allows DOS applications to run on the new operating system. Programs run in the DOS compatibility box are said to run in real mode (unprotected). OS/2 cannot run other tasks at the same time when a program is running in real mode.

DOS prompt

Typically the greater-than sign (>), preceded by a letter that designates the default drive.

DOS shell

A software program that enhances DOS by adding additional functions, such as a menu generator, path utility, file/disk backup utility, and extended DOS features.

Double-density

These disks include 40 tracks per side, which is double the number on the obsolete single-density diskettes.

Double-sided

A diskette that can store data on both surfaces. The alternative is a single-sided diskette. Double-sided media obviously store more data.

Drive designation

Indicates the drive on which a file is located.

Emulation

The process whereby one workstation looks and behaves like another; i.e., an IBM PC can emulate a Burroughs A10, IBM 4381, or other mainframe computer.

End-of-file mark — A character in an ASCII text file generated by pressing the <F6> Key or <CTRL-Z>. This special character marks the end of the file on disk.

Enterprise — Any operation about which data is generated and stored on a computer system. An enterprise may be a person, department, company, state, institution, or similar entity.

Expanded memory — A specification for accessing more than 1MB of memory using a bank switching scheme.

Extended memory — Available in machines using the Intel 80286 microchip, which can address memory above the 1MB barrier imposed by older Intel 8088 or 8086 machines. The amount of memory that may be simultaneously addressed is "extended" to 16MB.

Extension — A portion of the complete filename that follows the file specification and a delimiter. Can be up to three characters long.

External DOS commands — DOS commands that must be loaded from disk each time they are executed.

External drive — A disk drive or hard disk not housed with the PC's system unit.

FAT — The file allocation table (FAT) stores information that indicates the sectors assigned to each file on a diskette or hard disk. The FAT also indicates the sectors on a disk that are free and may be used to store new data or new files.

File control block — Stores data relative to a specific file used in a computing environment.

Filename — A group of characters that identifies a file saved to disk or tape.

File server — A device linked into a LAN, which is dedicated to storing public and private files for the network.

File-sharing — The ability to share files between multiple users.

File specification — Up to eight characters that identify a file.

Firmware

Software code on a microchip. ROM BASIC and ROM-BIOS are examples of firmware.

Fixed disk

Another term for a hard disk. A medium for storing data. Its storage capacity is measured in megabytes.

Floppy diskettes

A flexible, removable storage media for data. Floppy diskettes come in a variety of sizes and capacities. DOS 4.x and previous versions of DOS support diskettes ranging from 160K to 1.44MB.

Footprint

The space on a desk, table, or shelf occupied by a PC or workstation and all related peripherals.

Foreground task

In the DOS environment, the task with high priority is referred to as the foreground task. When competition arises between the foreground and background tasks over access to the CPU, the higher-priority task wins. Foreground tasks always win over background tasks.

Formatted

Indicates that the storage medium (hard disk or floppy diskette) has been initialized and prepared to store data.

Fragmented

The condition in which allocation units (portions) of a file are scattered in non-contiguous areas on a disk resulting in more time being required to read the file into memory.

Function key

One of ten (or twelve) keys labeled as F1 through F10 (or F12 on some keyboards). Function keys perform special assigned functions in application programs or at the DOS prompt.

Gateway

An access point from which a PC or network is linked to another network, mainframe, or data base.

Global wildcard characters

The question mark and asterisk which, respectively, can substitute for one or several characters in a filename.

Hard disk

A device used to store data. It is composed of several platters with recording surfaces coated with magnetic oxide. The platters are similar to a diskette except they are rigid and store more data. A 10MB hard disk can store data that would otherwise require 25 to 30 360K floppy diskettes. The hard disk is non-removable storage.

Hard-disk crash

Typically, the failure of a hard-disk subsystem caused by the read-write head coming in destructive contact with one of the platters on which data is stored.

High-density

Contains the same number of tracks (80) as quad-density disks, but packs more sectors per track (see **Quad-density**).

Home position

The upper left corner of the monitor screen. Technically, row 1, column 1 of the screen display.

Immediate mode

The situation in which DOS commands are invoked and executed immediately.

Initialization

The process by which diskettes are prepared to store data. On a double-sided, double-density floppy disk, 40 electronic tracks are recorded to the disk and divided into sectors. IBM PC DOS and MS-DOS use the FORMAT command to initialize a disk.

Initialize

The process used by DOS to prepare diskettes to store data (see **Initialization**).

Input

The process of entering data into the computer.

Internal DOS commands

DOS commands that are resident in memory as soon as the system unit is booted.

Kilobyte

A measure of storage on a disk. One kilobyte includes 1024 bytes of data and is referred to as 1K (shorthand for kilobyte). 1K is technically 2 to the power of 10.

Local area network (LAN)

A system in which multiple workstations are connected through cabling or telecommunications in order to provide file-sharing.

Logical drive

The name by which a user refers to a drive. The drive may physically be drive C:, but to the user it may be logical drive D:.

Look-ahead buffers

Allows DOS to read more than the current cluster or allocation unit called for by an application program. The requested sectors are read plus the subsequent clusters assigned to the file. Speeds up file access and processing.

Lost clusters

Clusters or allocation units that are marked as used but not linked to a file in the FAT.

Macro

A collection of keystrokes that can be invoked by a single keystroke or quick key combination.

Manual backup systems

Manual, non-computing modes of conducting business, e.g., paper-and-pencil operations.

Media

Types of cabling used on a local area network. Could also refer to types of disk or tape storage.

Medium

Material used for storage and communication. In computer terms, the medium might be any of the following: floppy diskette, hard disk, tape, or types of cabling.

Megabyte

A measure of memory. One byte equals one character, one kilobyte equals 1024 bytes, and it takes 1024 kilobytes to equal 1MB (in the simplest terms, 1,048,576 characters).

Megahertz

A measure of speed. The speed of a PC is measured in terms of the speed of the computer's central processing unit. The original IBM PC operated at 4.77 megahertz (MHz), while the original IBM AT performed at 6 MHz. AT compatibles typically run at clock speeds between 10 and 12 MHz. The newer PS/2 computers run anywhere from 8 MHz (Models 25 and 30) to 25 MHz (Model 70-386).

Menu utilities

Network commands executed via a series of screens or menu options. Typically these accomplish the same tasks as command line utilities but are easier for novice users to execute.

Meta-string Text string that follows the dollar sign when invoking the PROMPT command. Each meta-string displays a specified text string or value as part of the DOS prompt.

Milliseconds The average hard-disk access speed is measured in milliseconds; that is, thousandths of a second.

Mnemonics Conventions used that are easy to remember or that use memory-assisting aids.

MS-DOS Microsoft disk operating system.

Multiprocessing The capability of a computer to handle several users at one time.

Multitasking The capability of a computer to handle several tasks at one time.

Network administrator Person charged with maintaining the network system including issuing passwords, providing technical support, etc.

Node One workstation or PC on a local area network system.

Non-printing characters Typically, control characters that are not meant to be printed but rather are interpreted by the computer and the printer to set document formats (see **Control characters**).

Operating System/2 IBM's successor to DOS; also referred to as OS/2. A new multitasking operating system announced in 1987 and delivered in 1988.

OS Operating system (see **DOS**).

Output The process of getting data out of the computer.

Parallel port An input/output port that can handle eight bits of data simultaneously.

Parameters Terms or items that modify a DOS command.

Parent directory

This term establishes the relationship between a directory and its subdirectories. Every directory except the root directory has a parent directory. Access to a subdirectory must be through its parent directory.

Parity

The use of an extra bit as an error checking strategy in each transmitted word. Depending on system settings, parity is either even or odd, or there is no parity. Because each transmitted word is composed of bits set to either on (1) or off (0), the parity bit is set in such a way as to assure that every word is even (for even parity) or odd (for odd parity). If a word is received that does not match the system's parity, a message is sent to retransmit the word or block of data.

Path

The route, beginning at the root directory, to a given file or program.

Pathname

The name specifying the path to a given file or program.

PC

A personal computer (PC) is a microcomputer or desk-top computer.

PC DOS

IBM personal computer disk operating system.

Peripherals

Devices attached to a computer that enable the input and output of data. Examples include printers, modems, plotters, and disk drives.

Personal System/2

The new family of IBM PCs introduced in April 1987. The PS/2 family includes several different machines, most using the Micro Channel Architecture (MCA) bus design. Included are the Models 25, 30, 50, 70, and others.

POST

Power-on self-test; the self-diagnostic procedure a computer goes through when initially booted.

Presentation Manager

The new graphical user interface introduced with OS/2 Standard Edition 1.1.

Print queue

A list of files to be printed. The queue is created by the PRINT command.

Private volume — A disk, directory, or subdirectory set aside for the exclusive use of a designated user on a local area network or multiuser system. Other users cannot access files on a private volume.

PRN — DOS reserved word used to indicate parallel port one; that is, LPT1:.

Protocol — The settings used for telecommunication of data. Protocol is set so the sending and receiving computers know what to expect in terms of word size, parity, and baud rate.

Public volume — A disk, directory, or subdirectory that may be accessed by any user on a local area network or multiuser system. Shared files and shared programs must reside on a public volume.

Quad-density — A high-density diskette that stores four times the number of tracks contained on a single-density disk. Quad-density disks contain 80 electronic tracks per side.

RAM — Random access memory; the memory used for data and program storage in a computer.

RAM drive — See **Virtual disk** and **Virtual drive.**

Read-and-write — The ability to view and modify data.

Read-only — The ability only to view data. The user is restricted from modifying the data.

Redundancy — A concept used in disaster recovery planning that requires the duplication of hardware and software and the periodic backup of data to ensure that the business of an enterprise may continue with minimal disruption.

Reserved characters — Characters used by DOS for special purposes. These characters may not be used in filenames.

Reserved words — Words used by DOS for special purposes. These words may not be used for filenames, including batch filenames.

ROM — Read-only memory; the memory stored on computer hardware (microchips) that cannot be deleted or altered by stored program instructions. The user cannot write data or instructions to ROM, though it is part of the PC's RAM.

Root directory — The main directory created by DOS every time a disk is formatted. Directories created by the user are called subdirectories or directories under the root directory.

Secondary storage — Memory storage space located external to the CPU and its memory chips (tape backup, hard disk, floppy disk). The term *storage* is synonymous with memory.

Sector — A block of storage space on a disk. A sector can store 512 bytes. The number of sectors per track varies from 8 to 18 depending on the disk medium.

Serial port — An input/output port that can transmit only one bit of data at a time.

Single-density — An obsolete recording density that allows only 20 tracks to be recorded on each side of a diskette.

Single-sided — A diskette that can only store data on one side (one recording surface).

Source disk — The disk on which data to be copied, moved, or edited is found.

Standard devices — Peripherals designated by DOS using reserved words. Standard devices include the console and parallel and serial ports.

Stopbit — A bit sent to signal the end of a transmitted word, frame, or block of data.

Storage — Another term for memory. RAM and ROM are types of internal storage, while diskettes and hard disks are types of external storage.

Subdirectory — A directory created by the user either under the root directory or under another user-created directory. Each subdirectory stores files designated by the user. Directories and subdirectories are created to help organize files on a hard disk.

Superstation — The primary workstation on a disk-server-based network system. The superstation is typically used as a normal workstation in addition to servicing as the network administrator's console for network configuration and management.

Switch — Another term for a parameter (see **Parameters**).

Syntax — The structure of a language. The form in which commands and parameters are executed so the computer can understand what action should be taken.

System prompt — See **DOS prompt**.

Target disk — The disk that is the recipient of a copy, move, or modify operation. Sometimes referred to as the destination drive.

Terminal software program — A software program that facilitates the transfer of data through telecommunications. Typically these programs are menu-driven and provide an easy user interface for setting the telecommunications protocol, as well as storing commonly dialed phone numbers of on-line data bases and bulletin boards.

Text file — Often referred to as an ASCII text file or ASCII file. This file contains text, i.e., letters of the alphabet, punctuation, and numbers. The TYPE command will display the contents of a text file. Other types of files contain numeric values representing text or data and are not readable using the TYPE command.

Time-out — If a device does not respond to the CPU within a given period of time, a time-out error is generated. It means that a preset period of time has elapsed and control is returned to the CPU.

Toggle keys

Keys, such as <CAPS LOCK>, which change the action of the keyboard. Press it once, and all alphabetical characters are displayed in capitals. Press <CAPS LOCK> again, and lowercase letters are generated. Other toggle keys include <NUM LOCK> and <SCROLL LOCK>.

Topology

The pattern by which workstations are connected to a LAN. Two popular options are the star and the bus topology.

Tracks

Magnetic paths recorded on the magnetic oxide surface of a diskette. Data is stored on tracks with each track broken down into sectors.

Unformatted

A disk or storage medium is consider unformatted until its surface is prepared to store data.

Upward compatibility

The assurance that new developments in hardware and software may be used by an older machine.

User-friendly

A term referring to the ease with which the user interfaces with the computer.

User interface

The screen displays and commands that facilitate interaction between the user and the machine.

Virtual disk

A simulated disk. A portion of RAM is set aside to store data as if it were a floppy disk drive. Data stored in a virtual disk are volatile (see **Virtual drive**).

Virtual drive

A storage location such as a disk drive, which resides in an allocated part of memory. Data stored on a virtual drive are more volatile, but have a faster access time than when stored on a regular disk drive (see **Virtual disk**).

Warm boot

The action whereby the computer is electronically turned off and then back on. Caused by pressing <CTRL-ALT-DEL> on an IBM PC or compatible.

Wildcard characters

The asterisk and/or question mark, which may substitute for one or more characters in a filename used as part of a DOS command. Has the same meaning as global wildcard characters.

Word size — See **Data bits.**

Workstation — A terminal and keyboard at which a user may compute. Some workstations are PCs while others are dumb terminals connected to a mainframe computer.

Workstation prompt — The command line on a network workstation. Corresponds to a DOS or system prompt on a stand-alone PC.

Write-protect — The ability only to view data (see **Read-only**).

APPENDIX B

FDISK AND HARD DISK FORMATTING

Before you can use your hard disk, it must be prepared to receive DOS. The FDISK program will allow you to create a partition on the hard disk for DOS. A partition is an area of the hard disk where you can place the DOS operating system.

In this hard-disk setup you are going to place DOS on a 10-megabyte hard disk which does not have a DOS partition. If you want to create a partition on a hard disk with a partition you must delete the old partition, before you create the new partition.

STARTING THE COMPUTER

1. Insert a DOS 3.3 diskette into drive A:.

2. Turn on the computer.

3. Enter the current date and time.

FDISK PROGRAM

1. Type **FDISK**

2. Press <ENTER>.

The screen will display the response for the FDISK command. The FDISK command will only give you a list of four actions to choose from when you have one hard disk in your computer.

```
IBM Personal Computer
Fixed Disk Setup Program Version 3.30
      (C)Copyright IBM Corp. 1983, 1987


FDISK Options

Current Fixed Disk Drive:1

Choose one of the following:

     1.      Create DOS Partition
     2.      Change Active Partition
     3.      Delete DOS Partition
     4.      Display DOS Partition.

Enter choice: [1]

Press ESC to return to DOS.
```

3. Press <ENTER> to create the DOS partition.

The screen will display the following response:

```
Create DOS Partition

Current Fixed Disk Drive:1

     1.      Create Primary DOS partition
     2.      Create Extended DOS partition

Enter choice: [1]

Press ESC to return to FDISK options
```

4. Press <ENTER> to create the Primary DOS partition.

The screen will display the following response.

```
Create Primary DOS Partition

Current Fixed Disk Drive:1

Do you wish to use the maximum size
for a DOS Partition and make the DOS
Partition active (Y/N)?      [Y]
```

5. Press <ENTER> for the maximum size for the DOS partition.

The screen will display the following response:

```
System will now restart


Insert DOS diskette in drive A:
Press any key when ready.....
```

6. Press <ENTER>.

7. Enter the current date.

8. Press <ENTER>.

9. Enter the current time.

10. Press <ENTER>.

FORMAT FIXED DISK

The hard disk is now prepared to receive DOS. The FORMAT command will place the DOS operating system onto the partition you created with the FDISK command.

1. Type **`FORMAT C:/S/V`**

2. Press <ENTER>.

The screen will display the following response for the FORMAT command.

```
A>FORMAT C:/S/V
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format(Y/N)?_
```

3. Type **Y**

4. Press <ENTER>.

The screen will display the following message on completion of the FORMAT command.

```
A>FORMAT C:/S/V
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format(Y/N)?Y

Format complete
System transferred

Volume label (11 characters, ENTER for none)?_
```

5. Type **Hard Disk** for the volume label.

6. Press <ENTER>.

The screen will display the following message:

```
10585088 bytes total disk space
   86016 bytes used by the system
10499072 bytes available on disk

A>
_
```

7. Remove the DOS diskette from drive A:.

8. Press <CTRL-ALT-DEL> to restart the computer.

APPENDIX C

TYPES OF DISK DRIVES

The variety of disk drives for IBM-compatible personal computers seems to increase each year. It may be helpful to review the types available. The major types of disk drives currently on the market are

180K disk drive This drive was issued with the original IBM PC. It could read only single-sided (SS), single-density (SD) diskettes. Until DOS 2.0 was released, this drive could format a disk to hold only 160K of data. This drive is generally obsolete.

360K disk drive This drive reads a double-sided (DS), double-density (DD) 5¼-in. diskette and is common on the IBM PC and XT, and as the second drive (B:) on the AT. Most PC compatibles include at least one 360K drive unless they are AT compatibles (which typically use a 1.2MB floppy as drive A:). If you are working with a PS/2, you probably do not have access to a 360K drive unless it is hooked up as an external drive.

720K disk drive This drive reads a 3½-in. diskette and is included in the PS/2 Models 25 and 30. Some PCs may have the 720K drive attached as an external drive. The 3½-in. drive uses a 1MB diskette that stores 720K after it is formatted.

1.2MB disk drive The 1.2MB disk drive is most often found in the IBM AT and AT compatibles. It reads a 5¼-in. double-sided (DS), quad-density (QD) diskette.

1.44MB disk drive — The new IBM PS/2 computers (Models 30-286, 50, 50Z, 60, 70-386, and 80-386) each come equipped with an internal 3½-in. disk drive. Its storage capacity is 1.44MB. The 3½-in. drive uses a 2MB diskette that stores 1.44MB after it is formatted.

The very first IBM PC disk drives only read SS 5¼-in. 160K diskettes. With the advent of DOS 1.1, drives were introduced that could read DS 5¼-in. 320K diskettes or the SS diskettes. In essence, there were two types of drives: those that could read one side of a disk and those that could read both sides. DOS 2.0 worked with the same 5¼-in. drives, but increased the amount of storage per disk. The disk drives capable of reading only SS diskettes have largely disappeared. The drives capable of reading a DS 5¼-in. diskette are still the mainstay of business and education today, though they are losing ground to the new 3½-in. disk drives because of their increased storage capacity.

APPENDIX D

A CONCISE HISTORY OF MS-DOS/PC DOS

PC DOS was developed by Microsoft Corporation as an operating system for IBM. The original version (DOS 1.0) supported floppy drive systems only. When the IBM PC was introduced in the spring of 1981, only floppy-based systems were available. The operating system developed by Microsoft for IBM is technically referred to as IBM DOS. PC DOS doesn't exist, except as a nickname assigned by users. As soon as PC DOS 1.0 became available, a similar product was simultaneously offered by Microsoft. It was licensed to other vendors and the general public as MS-DOS.

By the time hard-disk drives were available for the PC XT, DOS 2.0 was available. It provided additional commands which supported the use of hard-drive systems. In early 1986, DOS 3.0 became available to add yet another dimension to the IBM family of computers. DOS 3.0 was the first version to support the networking of several microcomputers. Networking is simply the process of linking several microcomputers together so that files and data may be shared. A comparison of features and fixes offered by each version of DOS is provided on page D-3.

Late in March 1986, IBM announced the release of DOS 3.20. Despite several new commands, many users considered 3.2 a minor release and took a wait-and-see approach. The new enhancements included the addition of the XCOPY, REPLACE, EXDSKBIO.COM, and DRIVER.SYS commands. At the time, several commands--FORMAT.EXE, DISKCOMP, DISKCOPY, ATTRIB, COMMAND, SHELL, and SELECT--were enhanced. Of these enhanced commands, three were significant to users. First, DISKCOPY and DISKCOMP added support for the new double-sided 3½-in. 720K diskette. Second, the new improved FORMAT command would not let users format the default drive unless the drive letter was specified. DOS 3.20 also provided additional support for the IBM Token-Ring Network.

When DOS 3.3 appeared, several new commands were added, including APPEND, FASTOPEN, NLSFUNC, and CHCP. Of significance to users were the enhanced ATTRIB, BACKUP, RESTORE, FDISK, DATE, and TIME commands. BACKUP no longer required users to format diskettes before beginning a backup. ATTRIB gave the user greater control over changing the attributes of a file or group of files. Further, FDISK allowed the user to partition one large real hard disk into several logical drives.

By the time DOS 3.3 was widely accepted, users began anticipating the next evolutionary step in OS development. As software programs became more and more powerful and full-featured, PC users looked for a "breakthrough" DOS capable of two new monumental enhancements. First is the ability of the OS to address more than 640K of memory. The 640K barrier limited the size and complexity of programs that could be written for the PC. Second, the new DOS was expected to be the first true multitasking IBM micro-based operating system. Multitasking is the capability of a micro to manage more than one activity or operation at a time. An example would be the capability to sort extensive records in the background (a part of memory set aside for such a function) while the user continues working with another application in the foreground.

Rather than taking a single path to an enhanced version of the operating system, IBM chose not to release a multitasking version of DOS and instead chose the dual route of DOS 4.x and OS/2.

In April 1987, IBM responded to speculation by introducing the PS/2 series of PCs and announcing the future release of a new operating system called Operating System/2 (OS/2). OS/2 addressed the needs of users by providing access to more than 640K of internal working memory. The ceiling was raised to allow access to 16MB of RAM. Further, OS/2 provided multitasking capabilities. The price of these new features was steep. OS/2 cannot run on the large installed base of 8088 and 8086 PCs (IBM PC and XT and PC clones). At a time when the cost of RAM chips was skyrocketing, OS/2 required an 80286 or 80386 processor (found in the PC AT and most of the PS/2 machines) plus a minimum of 2 to 4MB of RAM.

The initial announcement from IBM seemed to indicate that there would be no further development of DOS. However, acceptance of the PS/2 line and OS/2 was unexpectedly slow. IBM continued to lose market share to compatible makers who cranked out PC and AT clones for use with MS-DOS 3.3 and earlier versions. By August 1988, IBM released DOS 4.00.

Version 1.0

Did not support serial communication.

Had bugs in BASIC. Supported single-sided diskettes (8 sectors per track).

DATE and TIME commands were external.

Files were only date (not time) stamped.

Supported IBM PC.

Version 1.05

Fixed BASIC bugs.

Version 1.1

Mode command supported serial communications.

Supported double-sided diskettes (8 sectors per track).

DATE and TIME commands became internal.

Files were date and time stamped.

Verification and file concatenation added to COPY command.

EXE2BIN program added for programmers.

Officially retired by IBM in 1985.

Version 2.0

Hard-disk management commands added.

Supported IBM XT.

Version 2.1

Corrected bugs in 2.0.

Added support for PC jr.

Added support for half-height disk drives.

Version 3.0

Supports quad-density disks.

Can configure keyboard to country of use.

Can configure time and date display.

Supports display of characters with an ASCII value greater than 127.

Supports file-sharing and networking.

Supports RAM disk.

Supports IBM AT.

Version 3.1

Commands added to reassign or combine drives and directories.

LAN support enhanced plus added support for PC Token Ring Network.

Version 3.2

Support added for 3½-in. (720K) diskette.

Added safety features to FORMAT command.

Added REPLACE command to selectively replace files on a target disk with files of the same name on the source diskette.

Version 3.3

Added APPEND, FASTOPEN, CHCP, and NLSFUNC commands.

Added code page switching on an EGA, IBM PC Convertible LCD, or PS/2 display system.

DIFFERENCES BETWEEN PC DOS AND MS-DOS

Although the two versions of DOS used by IBM and manufacturers of IBM-compatible computers are basically the same, there are some differences. However, these differences continue to decrease.

MS-DOS is the generic product, whereas PC DOS has been customized to create a machine-specific product for the IBM line, including the following machines: IBM PC, IBM XT, IBM AT, and the PS/2 models. Originally, MS-DOS was considered the Chevy version, while PC DOS was the Cadillac. Both products would get you from "here" to "there," but earlier versions of PC DOS had some additional bells and whistles.

Beginning with MS-DOS version 2.11, Microsoft attempted to standardize both versions of DOS. PC DOS 2.1 and MS-DOS 2.11 were essentially the same. If a new version of PC DOS added features, then the subsequent version of MS-DOS would play catch-up.

By the time PC DOS and MS-DOS 3.3 were issued in April 1987, the two operating systems were virtually identical. With version 3.3, both operating systems added the FASTOPEN command and the ability to create an extended DOS partition on a hard disk (with more than 32MB of memory) and address it as a logical drive. Despite these small variations, most observers anticipate that any differences between MS-DOS and PC DOS are minimal or temporary, unless IBM decides to become more restrictive with its operating system.

LIST OF INTERNAL DOS COMMANDS

The following is a list of all internal DOS commands. Remember that these commands may be used whenever the DOS prompt appears. They are automatically loaded into memory when your system is initially booted.

DOS commands are printed in uppercase throughout this text. However, DOS commands are not case-sensitive. At the keyboard they may be entered in upper- or lowercase or a combination of the two.

In going from DOS 1.0 to 3.3, some changes in internal commands have been made. As a historical record, note the expansion of available DOS commands with each new version.

DOS 1.1	DOS 2.0	DOS 3.1	DOS 3.2	DOS 3.3
(BATCH)*	(BATCH)*	(BATCH)*	(BATCH)*	(BATCH)*
BREAK	BREAK	BREAK	BREAK	
BUFFERS	BUFFERS	BUFFERS	BUFFERS	
CHCP				
CHDIR	CHDIR	CHDIR	CHDIR	
	CLS	CLS	CLS	CLS
		COMMAND	COMMAND	COMMAND
COPY	COPY	COPY	COPY	COPY
		COUNTRY	COUNTRY	COUNTRY
		CTTY	CTTY	CTTY
	DATE**	DATE**	DATE**	DATE**
	DEL	DEL	DEL	DEL
		DEVICE	DEVICE	DEVICE
DIR	DIR	DIR	DIR	DIR
	ECHO	ECHO	ECHO	ECHO
ERASE	ERASE	ERASE	ERASE	ERASE
		FCBS	FCBS	FCBS
	FOR	FOR	FOR	FOR
	GOTO	GOTO	GOTO	GOTO
	IF	IF	IF	IF
			LASTDRIVE	LASTDRIVE
	MKDIR	MKDIR	MKDIR	MKDIR
	PATH	PATH	PATH	PATH
PAUSE	PAUSE	PAUSE	PAUSE	PAUSE
		PROMPT	PROMPT	PROMPT
REM	REM	REM	REM	REM
	REN	REN	REN	REN
RENAME	RENAME	RENAME	RENAME	RENAME
	RMDIR	RMDIR	RMDIR	RMDIR
	SET	SET	SET	SET
		SHELL	SHELL	SHELL
	SHIFT	SHIFT	SHIFT	SHIFT
				STACKS
				SWITCHES
	TIME**	TIME**	TIME**	TIME**
TYPE	TYPE	TYPE	TYPE	TYPE
	VER	VER	VER	VER
	VERIFY	VERIFY	VERIFY	VERIFY
	VOL	VOL	VOL	VOL

* Note: This command, unlike all others, is not executed by typing its name.

** Note: In DOS 1.0 these were external commands.

LIST OF EXTERNAL DOS COMMANDS

The following are external DOS commands. External DOS commands may be executed whenever the DOS prompt appears only if the corresponding DOS file is accessible on disk. These files are not automatically loaded into memory when the system is booted.

DOS 1.1	DOS 2.0	DOS 3.1	DOS 3.2	DOS 3.3
				APPEND
	ASSIGN	ASSIGN	ASSIGN	ASSIGN
		ATTRIB	ATTRIB	ATTRIB
	BACKUP	BACKUP	BACKUP	BACKUP
CHKDSK	CHKDSK	CHKDSK	CHKDSK	CHKDSK
COMP	COMP	COMP	COMP	COMP
	DEBUG	DEBUG	DEBUG	DEBUG
DISKCOMP	DISKCOMP	DISKCOMP	DISKCOMP	DISKCOMP
DISKCOPY	DISKCOPY	DISKCOPY	DISKCOPY	DISKCOPY
			DRIVER	DRIVER
EXE2BIN	EXE2BIN	EXE2BIN	EXE2BIN	EXE2BIN
	EDLIN	EDLIN	EDLIN	EDLIN
				FASTOPEN
	FDISK	FDISK	FDISK	FDISK
		FIND	FIND	FIND
FORMAT	FORMAT	FORMAT	FORMAT	FORMAT
		GRAFTABL	GRAFTABL	GRAFTABL
	GRAPHICS	GRAPHICS	GRAPHICS	GRAPHICS
		JOIN	JOIN	JOIN
		KEYB*	KEYB*	KEYB*
		LABEL	LABEL	LABEL
MODE	MODE	MODE	MODE	MODE
		MORE	MORE	MORE
				NLSFUNC
	PRINT	PRINT	PRINT	PRINT
	RECOVER	RECOVER	RECOVER	RECOVER
			REPLACE	REPLACE
	RESTORE	RESTORE	RESTORE	RESTORE
		SELECT	SELECT	SELECT
		SHARE	SHARE	SHARE
		SORT	SORT	SORT
		SUBST	SUBST	SUBST
SYS	SYS	SYS	SYS	SYS
	TREE	TREE	TREE	TREE
		VDISK	VDISK	VDISK
			XCOPY	XCOPY

* There are five variations of the KEYB command: KEYBFR, KEYBGR, KEYBIT, KEYBSP, and KEYBUK.

APPENDIX E

MS-DOS/PC DOS COMMAND REFERENCE

APPEND;	Resets the data path to NUL.
APPEND C:\WORD\WP	Opens a path to data files in the C:\WORD\WP directory.
ATTRIB +A Filename	Turns the specified file's archive bit on.
ATTRIB -A Filename	Turns the specified file's archive bit off.
ATTRIB +R Filename	Marks the specified file as read-only.
ATTRIB -R Filename	Marks the specified file as read-and-write accessible.
BACKUP C:*.* A:/M	Backs up all modified files (since the last backup) in the current directory of hard-disk C: to floppy disks in the A: drive.
BACKUP C:*.* A:/S	Backs up all files from all directories and subdirectories on hard-disk C: to floppy disks in the A: drive.
CD ..	Switches the default directory path from the current directory to its parent directory.
CD\	Switches the default directory path from the current directory to the root directory.

CD DOS	Switches the directory path to the DOS subdirectory if it exists under the current directory.
CD\DOS	Switches the directory path to the \DOS subdirectory.
CHKDSK A:	Provides a status report on the internal working memory and disk storage contained on the diskette in drive A:.
CHKDSK A:/F	Provides a status report on the internal working memory and disk storage contained on the diskette in drive A:, and fixes lost clusters by making them into individual files.
CLS	Clears the monitor screen and moves the DOS prompt and cursor to the home position (upper lefthand corner).
COMP A:*.* C:	Compares all files on the disk in drive A: with like-named files on drive C:.
COPY A:FILE.DOC B:	Copies the file FILE.DOC from the disk in drive A: to the disk in drive B:.
COPY A:FILE.DOC B:/V	Copies and verifies that the file FILE.DOC was accurately copied.
DATE	Prompts you to set or change the system date.
DEL A:*.*	Deletes all files from the disk in drive A:.
DEL A:*.*/P	Deletes all files from the disk in drive A: only after you respond "Y" to the prompt "Delete (Y/N)?" for each file.
DEL A:OLDFILE.DOC	Deletes the file named OLDFILE.DOC from the disk in drive A:.
DEVICE = RAMDRIVE.SYS 64 128 64	Creates an MS-DOS virtual drive when this command is added to the CONFIG.SYS file.

DEVICE = VDISK.SYS 64 128 64	Creates a PC-DOS virtual drive when this command is added to the CONFIG.SYS file.
DIR	Lists files on the current disk drive including filename, date and time created, size in bytes, total number of files, and total bytes free on the disk.
DIR|MORE	Generates a listing of all files on the default drive, one screenful at a time.
DIR/P	Causes the screen to stop scrolling after 23 lines on the current drive are viewed.
DIR >PRN	Redirects the output from the DIR command to a printer connected to parallel port one.
DIR|SORT	Generates an alphabetical listing of all files on the default drive.
DIR|SORT/R	Generates a listing of all files on the default drive in reverse alphabetical order.
DIR/W	Displays a horizontal directory of files on the current drive. Lists only filenames, total number of files, and bytes free on the disk.
DISKCOMP A: B:	Compares the disk in drive A: to the disk in drive B:. The comparison is made track by track and sector by sector.
DISKCOPY A: A:	Makes an exact duplicate in drive A: of the original (source) disk in drive A:. DOS will prompt you to insert the source diskette and then the target diskette at the appropriate times. The copy is made track by track and sector by sector.
DISKCOPY A: B:	Makes an exact duplicate in drive B: of the original (source) disk in drive A:. The copy is made track by track and sector by sector.

EDLIN NEWFILE.BAT	Creates or edits a batch file named NEWFILE.BAT.
FIND "Germany" A:COUNTRY.DOC	Locates all occurrences of the word "Germany" in the file COUNTRY.DOC on the disk in drive A:.
FIND/N "Germany" A:COUNTRY.DOC	Locates all occurrences of the word "Germany" in the file COUNTRY.DOC on the disk in drive A: and indicates the relative line number on which each occurrence appears.
FORMAT A:	Prepares a disk in drive A: to store data or programs. System files are not copied to the disk.
FORMAT A:/S	Prepares a self-booting system diskette to store data or programs.
MD DOS	Makes a new subdirectory named DOS under the current directory.
MODE BW40	Causes a color monitor to display a maximum of 40 characters per line across the screen in monochrome.
MODE BW80	Causes a color monitor to display a maximum of 80 characters per line across the screen in monochrome.
MODE CO40	Causes a color monitor to display a maximum of 40 characters per line across the screen in color.
MODE CO80	Causes a color monitor to display a maximum of 80 characters per line across the screen in color.
MODE COM1:12,E,7,1	Sets the serial port protocol to accept and transmit data at 1200 baud with even parity, seven data bits as the word length, and one stopbit.
MODE LPT1:	Resets redirected output back to parallel port one.

MODE LPT1:=COM1:	Redirects output from parallel port one to serial port one.
MODE LPT1:80,6	Causes output to a parallel printer to be printed 80 characters wide with six lines per vertical inch.
MODE LPT1:132,8	Causes output to a parallel printer to be printed 132 characters wide with eight lines per vertical inch.
PATH ;	Resets the path to search only the current directory for executable files.
PATH C:\DOS	Opens a path to executable files in the C:\DOS subdirectory.
PRINT	After creating a print queue, this command generates a status report of files in the queue and those being printed.
PRINT/D:PRN/B:10240/Q:5	Creates a print queue of 10,240 bytes (10K) that directs output to printer port one (PRN) and can hold a maximum of five files.
PRINT/T	Cancels the printing of all files in the print queue.
PROMPT	Resets the DOS prompt to show just the current drive and a greater-than sign.
PROMPT PG	Changes the DOS prompt to display the current path and drive and concludes the DOS prompt with the greater-than sign.
RD TRASH	Removes a subdirectory named TRASH that is under the current directory.
RD\TRASH	Removes a subdirectory named TRASH under the root directory.
RECOVER A:	Recovers data from any good sectors on the disk in drive A:.

RECOVER B:MESSEDUP.DOC	Recovers data from any good sectors storing portions of the file MESSEDUP.DOC on drive B:.
REN A:ONE.TXT TWO.TXT	Renames the file ONE.TXT (on the disk in drive A:) to TWO.TXT.
REPLACE A:*.* B:	Copies all files on A: to B: if the files already exist on B:.
REPLACE A:*.* B:/A	Copies all files on A: to B: if the files do not already exist on B:.
RESTORE A: C:*.*/S	Restores all files from the backup diskettes in drive A: to the appropriate directory or subdirectory on hard-disk C:.
RESTORE A: C:Filename	Restores the specified file from the backup disk in drive A: to the hard-disk C:.
SUBST	Indicates the existence of any virtual drives.
SUBST E: C:\WORD\WP	Creates a virtual drive E:. Files in the C:\WORD\WP directory may be accessed as if they existed on drive E:.
SUBST E:/D	Eliminates the virtual drive E:.
SYS A:	Copies the two hidden system files to a disk in drive A:, thus making the disk bootable.
TIME	Prompts you to set or change the current time.
TREE	Lists the current directory and all its subdirectories.
TREE/F	Lists the current directory, all its associated files, and all subdirectories under the current directory with their associated files.

TYPE FILEONE.TXT >PRN	Redirects the listing created by the TYPE command to a printer connected to parallel port one.
XCOPY A: B:	Copies all files from drive A: to B:.
XCOPY A:*.DOC B:	Copies all document files (with the .DOC extension) from A: to B:.

INDEX

.BAT
  order of execution 18
.COM
  order of execution 18
.EXE
  order of execution 18
<Back Space>
  erase incorrect characters 258
<PRTSC> Key 250
25 pin parallel connector 248
25 pin serial connector 248
3½-in. disk 23, 33
3½-in. disk drive 9, 32
5¼-in. disk 22
5¼-in. disk drive 32
640K
  maximum addressable memory 14
80286 microprocessor 7, 394
80386 microprocessor 394
8088 microprocessor 7, 14, 392

A

Allocation units 72
Alternative Processing strategies 190
American National Standards Institute 296
Analog signal 340
ANSI.SYS 296, 297, 300, 310, 318, 358
Archive attribute 215
Archive bit 214, 215
ASCII codes 312, 358
ASCII collating sequence 260, 263
ASCII value 260
Assembly language 371
ASSIGN command 202
Attached workstation 383
ATTRIB command 212, 215, 238
  boilerplate documents 214
  copied files 214
  read-and-write bit 213
  read-only files 214
  wildcards characters 213
  write-protected 213
Attribute bit 195
  read-only 213
Attribute byte 213
AUTOEXEC.BAT 119, 170, 179, 342

B

Background
  printing 330
Background colors 306
BACKUP command 193, 196, 198, 199, 203, 215, 222, 238
  complete or modified 191
  creates two files 198
  date parameter 195

BACKUP command (continued)
error codes 196
maxim 204
memory-resident programs 202
modified files 195
not interchangeable with COPY 199
number of diskettes 199
parameters 194
problem commands 202
redundancy 199
remove memory-resident programs 204
root directory 195
strategy 200
syntax 194
typical schedule 200
Backup schedule 199
Backup strategy
medium 191
BACKUP.LOG 196
Bad sectors 36, 37, 236
Bank switching 394
BASIC
stored in ROM 6
Batch Files 185
Baud rate 247, 340, 341
Binary digits 340
Binary files 356
Binary numbers 358
BIOS 5
Bits per second 340
Boilerplate 214
Boot record 192
Booting your system 6
BREAK 182
BUFFERS 182, 183, 295
Business software
types 4

C

Cabling 375
CD command 112, 115, 128
Central processing unit 4, 5
CHCP command 293, 318
CHDIR command 112, 115
CHKDSK command 58, 70, 71, 72, 77
CHKDSK/V command
with MORE filter 263
Clock ticks 327
CLS command 58, 59
Code page font file 295
Code page switching 288, 292, 294
Code pages 290, 292, 293, 294
Cold boot 34
Collating sequence 261
Color/graphics display 290
Command line utilities 383
COMMAND.COM 14, 36, 41
processes commands 14
COMP command 80, 95, 106
CON
default I/O device 249, 250
Concentric magnetic tracks 34
CONFIG.BAK 344
CONFIG.SYS 182, 295, 293, 296, 342, 344, 376, 378, 380
LASTDRIVE 380
Console 293, 295
monitor and keyboard 246
Conventional memory 343, 392
COPY command 81, 84, 89, 90, 92, 94, 95, 103, 106, 192, 193
attribute bit 214
back up limitations 192
back up strategy 193
COPY CON 82
command 246
Copy-protection 202
limited installs 203
methods 202
Copy-protection schemes 203
COUNTRY 182
Country code 292, 293
Creation time 195
Current directory 207

D

Data bits 340
Data communications 340
Data sharing
LAN 374
DATE command 58, 63, 77
DEBUG command 356, 358, 359, 363
COMPARE command 371
DUMP subcommand 362
ENTER subcommand 370

DEBUG command (continued)
LOAD command 369
MOVE command 371
NAME command 369
REGISTER command 372
SEARCH command 371
DEBUG prompt 359
DEBUG subcommands 361
Decimal numbers 356
Default device 246
Default drive 19, 119
DEL command 102, 106
Delimiters 15
DEVICE command 182, 318, 342, 351
virtual drive 342
Device driver 295, 296
Diagnostic disk
park head 191
DIR command 14, 39, 58, 67, 69, 77, 250
/P parameter 263
column numbers 262
Directory table 192, 199
Disaster recovery plan 190, 196
backups 196
impact on enterprise 190
periodic backups 190
Disk drives
three types 342
Disk handling guidelines 24
Disk platters 191
Disk sector 356
Disk space 70
Disk-to-tape backups
not supported 191
DISKCOMP command 51
DISKCOPY command 49, 192
not for hard disk back up 192
Diskettes
damage 21
data storage capacity 192
directory table entries 192
envelope 21
jacket 21
tools of computing 21
window 21
Display adapter 289
Display attributes 306
DOS 5
basic I/O control 5
file maintenance activities 10
manager of data transfer 10
not case-sensitive 13
subset of OS 5
DOS 2.x
back up hard disk only 191
BACKUP flaw 209
piping 259
PRINT command 326
DOS 3.x
ATTRIB added 213
enhanced BACKUP 191
network support 376
piping 259
DOS commands
available at DOS prompt 14
DOS prompt 7, 14, 60, 62
Double-sided diskette 112
Drive designation 15, 18
default drive 19
formal name 18
Drive designator 36, 379

E

ECHO command 173
EDLIN command 136, 144, 152, 166, 214, 356
COPY command 154
create text file 255
DELETE command 143
INSERT command 137
LIST command 141
MOVE command 156
QUIT command 146
REPLACE command 162
SEARCH command 159, 160
EDLIN prompt 146, 359
Electronic mail system 376
Emulation 392
End-of-file mark 82, 206
Enterprise 190, 375
ERASE command 102
Error checking 340
Error code 229
Error message 14
Escape sequences 299, 306

Exit codes 196, 198
Expanded memory 343, 392, 393, 394
Extended ASCII character set 310
Extended memory 295, 343, 392, 393, 394
  virtual drive 343
Extension 15
  1, 2, or 3 characters 17
  assigned by DOS 18
  automatically added 17
  DOS standards 18
External DOS commands 14, 80, 96
  .COM or .EXE extension 14

F

FAT (see file allocation table)
FCBS command 182, 376, 377, 388
  file control blocks 377
  guidelines 378
  least recently used 377
  open files 377
File allocation table 36, 72, 192, 385
File control block 376
File server 382, 385
  LAN 375
File specification 15
File-sharing
  locks 376
Filename 15
  complete 15, 20
  descriptive 16
  four parts 15
  legal characters 15
  spacing 13
Filename extensions 14
FILES command 182, 183, 295
Filters
  FIND 271
  modify data 259
  MORE 263
  SORT 260
Financial applications 202
FIND command 282
  cannot use wildcards 273
  case-insensitive 272
  count parameter 274
  locate all occurrences 271
  multiple file search 272
FIND command (continued)
  optional parameters 273
  parameters 273
  relative line number 273
  reverse switch 274
FIND filter 271
Firmware 7
Footprint 375
Foreground
  printing 330
Foreground colors 306
FORMAT command 36, 37, 53, 194
  redirect keystrokes 255
Future of IBM PC DOS
  references 397

G

Gateway 376
Global characters 20

H

Halfheight drives 8
Hard disk 7, 32, 113
  failure 191
  non-removable 33
  read-write head 191
  track zero error 191
Hard-disk crash 191
Hard-disk failure
  causes 192
Head access slot 33
Hewlett Packard LaserJet Series II 324
Hex values 357
Hexadecimal numbers 356
Hidden directory 203
Hidden files 14, 45, 192
Hidden system files 53
High density diskette 33
High density drive 33
High-capacity disk 36

I

IBM Diagnostic diskette 191
IBM PC
  revision B 14

IBM PC setup
typical 8
IBMBIO.COM 14, 36, 41
IBMDOS.COM 14, 36, 41
Icons 396
IF ERRORLEVEL command 196, 198, 229
Interface card 381
Internal DOS commands 14, 38, 75, 80, 97, 119, 393
resident in RAM 14
Internal print buffers 395
Interrupts 343
Inverse video 306
IO.SYS 14, 36

J

JOIN command 202

K

Key diskette 203
Key redefinition 310
Keyboard code 292
Keyboard designation 291

L

LABEL command 47
LAN 374
data sharing 374
disk server 375
FCBS command 377
file server 375
gateway 376
local area network 374
nodes 376
shared peripherals 375
wide area network 374
Laser printer 375
LASTDRIVE command 182, 376, 378, 380, 388
default value 378
maximum on system 378
syntax 380
LIM EMS 394
Local area network 374
no uniform standard 374
private directories 378
public directories 378
Local drives 382, 383
Locks 376
Login script 385
Lost clusters 72
Lost files 371
Low density drive 33

M

Magnetic oxide surface 22
MD command 112, 113, 128
Megabyte 392
Memory
addressed by Intel 80286 393
addressed by Intel 8088 393
expanded 392
extended 392, 393
Memory available 70
Memory expansion boards 394
Memory-resident programs 202, 392
Menu utilities 383
Meta-strings 60
Method of interface
serial or parallel 246
Microprocessor 10
MKDIR command 112, 113
MODE command 288, 289, 292, 293, 294, 295, 318, 324, 335, 340, 341, 351
40 or 80 column display 290
cancel redirection 325
condensed print 325
display adapters 289
five uses 288
parallel to serial switch 325
retry parameter 341
screen display shift 290
SELECT option 293
serial protocol 340
Modem 247
MORE filter 263, 282
Mouse 396
MS-DOS
DOS for compatibles 5
history 26
Microsoft operating system 5
MSDOS.SYS 14, 36

Multiprocessing 392, 395, 396
concurrent processing 395
multiuser systems and networks 395
Multitasking 395
Multiuser Systems
drawbacks 395

N

National language support 290
Native mode 394
Network administrator 376, 383
Network cards 381
Network commands
FCBS 376
LASTDRIVE 376
SHARE 376
Network drives 382, 383
Network printer 386
Network software 381
NLSFUNC command 293
Nodes 376
Non-English language characters 290
Non-system diskette 36, 37, 39
Novell 381
Novell network 382
NUL 249
diagnostic device 249

O

Operating system 5
Operating System/2 10, 396
OS/2 (see Operating System/2)

P

Parallel communications port 246
Parallel port 246, 294, 325
Parameters 12
Parent directory 207
Parity 340
Partial restore 210
PATH command 19, 112, 119, 122, 128, 237
Path name 15, 116
PAUSE command 177, 179
PC DOS
history 26
IBM Personal Computer operating system 5
Periodic backup 195, 204
Peripherals
shared 375
Personal System/2 9
Pipes
connect DOS commands 259
different from redirection 259
Piping
temporary files 259
POST (see Power-on self-test)
Power surges 192
Power-on self-test 6, 392
included in ROM 6
part of OS 6
Presentation Manager 396
PRINT command 327, 329, 330, 335
active queue files 328
interference 329
multitasking 326
optional parameters 326
Print spoolers 295
Printer port numbers 324
Private volume 378
PRN
alias for LPT1 250
default list device 328
PROMPT command 35, 58, 60, 296, 297, 298, 299, 300, 310, 318, 358
Protected mode 394
Protocol 340, 341
PS/2 models 10
Public volume 378

Q

Queue 326, 327, 328, 329
Radio frequency interference 192

R

RAM 6, 14
RAM disk
LASTDRIVE command 378
RAMDRIVE.SYS 295, 342, 344
default settings 343

RD command 112, 124, 128
Read-only attribute 214
  cannot erase 214
Read/write head 33, 191
Real mode 394
Reconstruct a file 237
RECOVER command 190, 236, 237, 238
Redefine keys 296
Redirecting output 329
Redirection 250, 282
  append to existing file 251
  channels output to devices 259
  input for DOS command 255
  overwrites old file 251
  to NUL 249
  to PRN 261
Redundancy 199
REN command 97, 106
REPLACE command 228, 229, 235, 238
Reserved characters
  DOS rules 16
Reserved words 248
RESTORE command 193, 196, 197, 198, 199, 203, 208, 222, 238
  error codes 198
  optional parameters 197
  partial 209
  recreate subdirectories 209
  single file 209
  subdirectories 209
  syntax 196
  with wildcard characters 208
RMDIR command 112
ROM 5
Root directory 20, 38, 62, 112, 113, 179, 195
  entries 192

S

Scan codes 310, 311, 358
Screen colors 306
Screen dump 37
Sectors
  per track 192
Security 190, 375
SELECT command 291
Self-booting system diskette 37, 41, 53
Serial port 246, 247, 294, 324, 325
  COM1 341
  COM2 341
Serial printers 247
Serial transmissions 247
SHARE command 376, 388
SHELL 182
Software
  network versions 375
SORT command 259, 267, 282
  ASCII collating sequence 260
  DOS 2.x versus 3.x 261
  filter 259, 260, 282
  parameters 262
  reverse order 262
Source disk
  original 13
Source file 12
Standard devices 246, 248
  aliases 248
  console 246
  default devices 246
  NUL 249
  reserved words 248
Static electricity 192
Stopbits 340, 341
Subdirectories 19, 112, 113, 123, 194, 209, 386
  restore 209
SUBST command 202
Switches 12
Syntax 11, 12
  CP/M different 13
  order of language 11
  specified order 12
Syntax diagram 13
  standard abbreviations 13
SYS command 45, 53
System date 63
System time 66

T

Tape backup systems 203
Target disk
  recipient of action 13
Telecommunications 341
  terminal software 341
Terminal program 341

Terminate batch job 175
Terminating codes 299, 310
Test pattern 290
Text editor 136
Text files 356
TIME command 58, 66, 77
Time slice 327
Time-out error 324
Topology 375
Track zero error 191
Transactional files 385
TREE command 112, 123, 128
Trustee rights 385
TYPE command 75, 100, 106, 196, 206, 214, 282
  with MORE filter 263

U

Upward compatibility 392, 396
  DOS 3.x compatibility box 396
User-friendliness 392
User-Friendly
  graphic interface 396

V

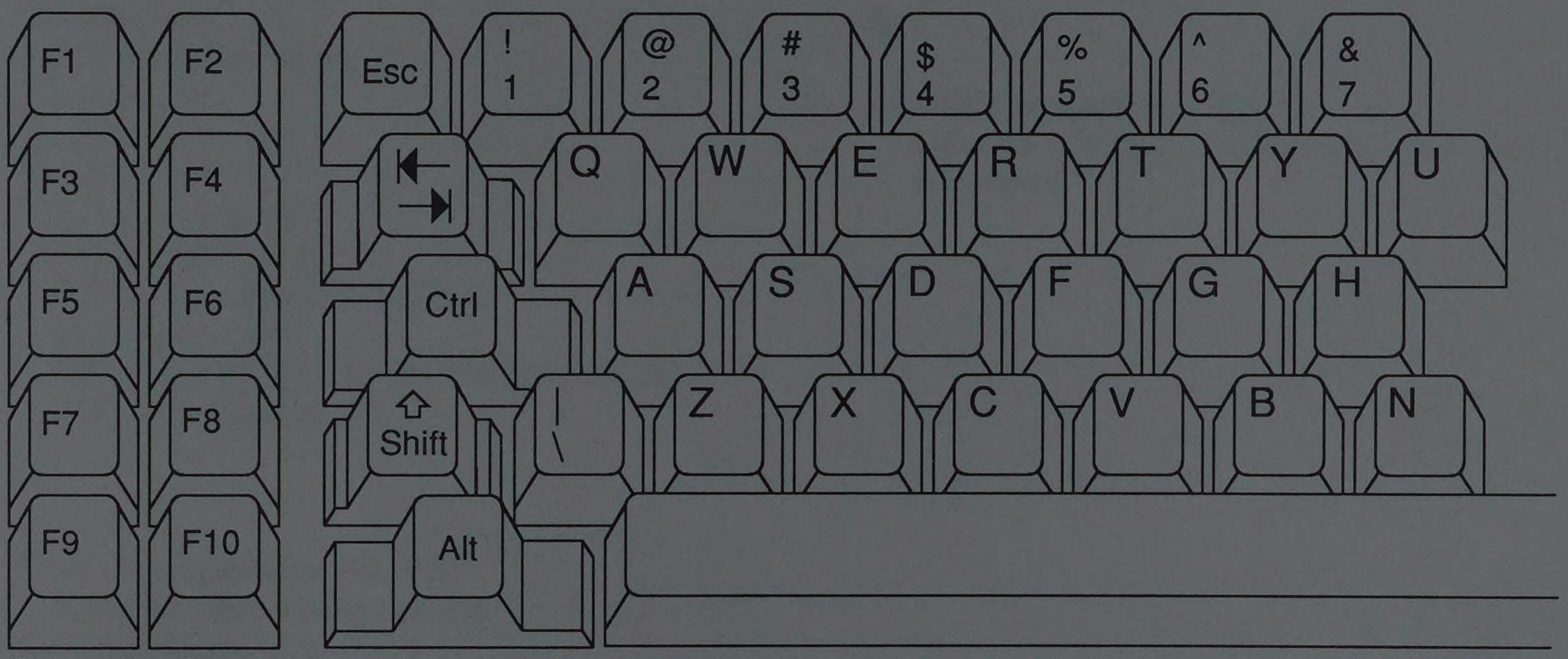
VDISK.SYS 295, 344
  default settings 343
  DOS virtual drive 342
  speed 342
Virtual drive 344
  CONFIG.SYS 342
  no formatting 342
  portion of RAM 342
  volatile memory 342
Volume label 39, 53, 192

W

Warm boot 34
Wildcard characters 21, 88, 90, 91, 103, 193, 209, 215,
    223, 328
  * or ? 20
  substitute for characters 20
Word length
  data bit size 341
Word sizes 340
Write-protect(ion) 21, 22, 213
  ATTRIB 213
Write-protect window or notch 22, 33

X

XCOPY command 223, 228, 239
  cannot copy hidden, read-only, or system files 223
  parameters 223

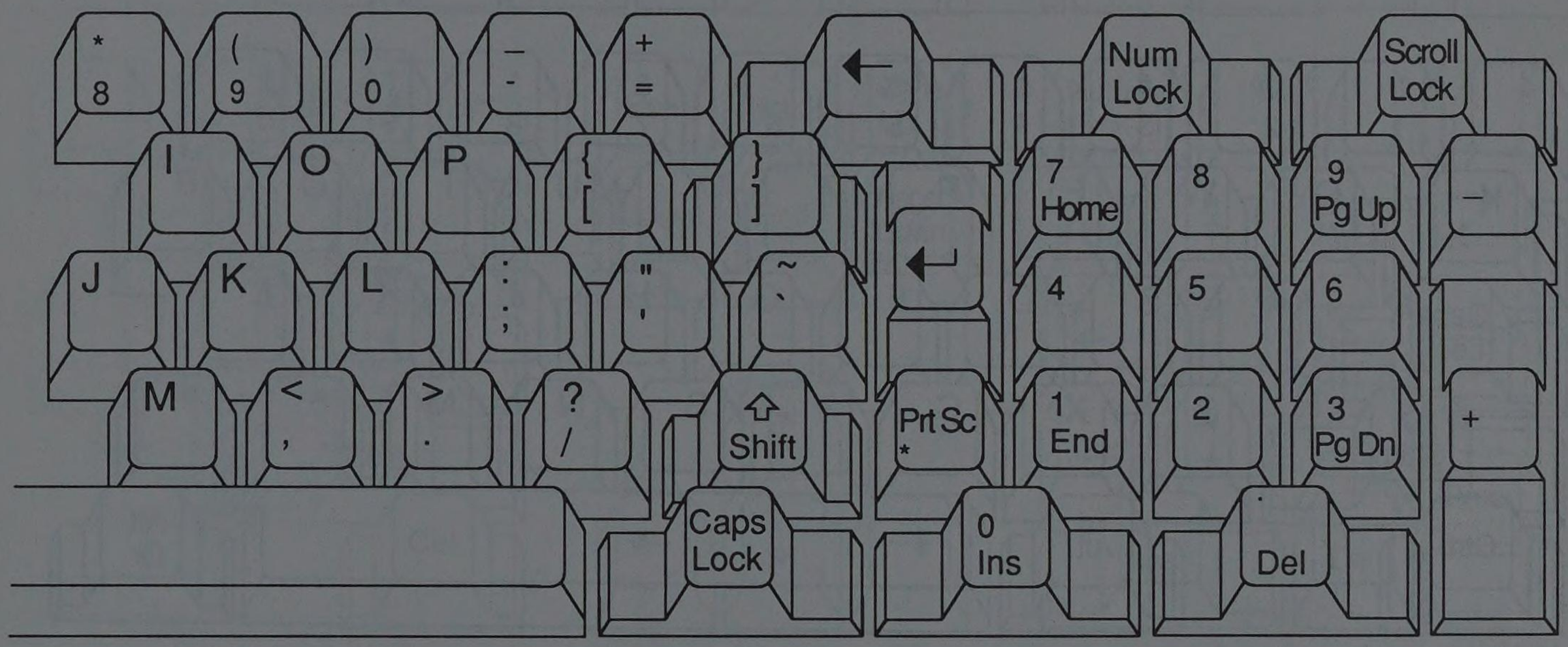


Standard Keyboard

DOS Keyboard

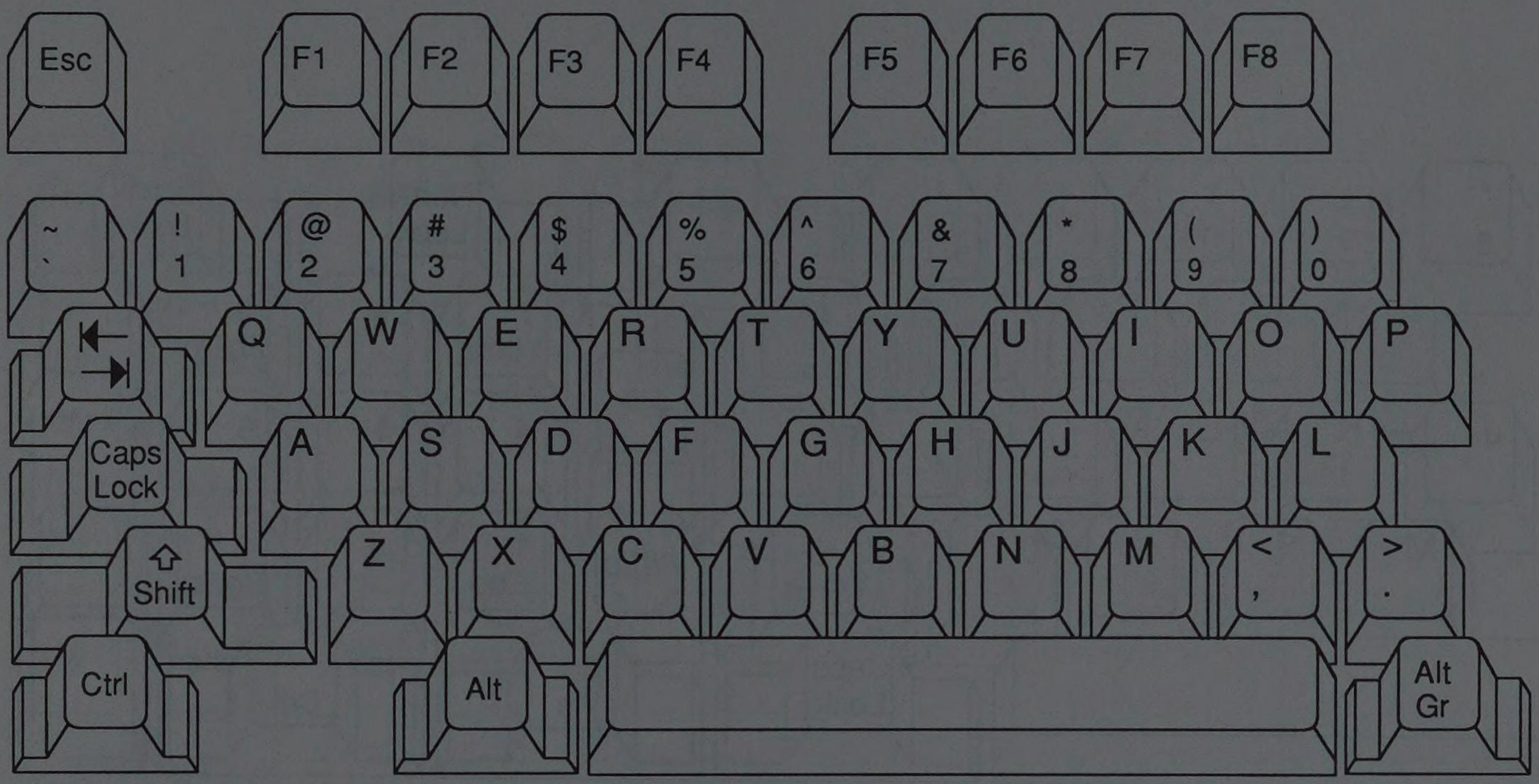
The last command entered at the DOS prompt is stored in a template buffer.

DEL	Does not copy the current template position to the command line
ESC	Voids the new command line and leaves template unchanged
F1	Plays back one character at a time from the template
F2	Plays back all characters from the template up to a specified character
F3	Plays back the remaining characters in the template
F4	Skips over all characters in the template up to a specified character
F5	Causes the current line to be the new template
F6	Inserts an End-of-file mark at the cursor location
Ins	Causes command line to toggle between insert mode (from keyboard) and insert mode (from the template)



IBM PC Abbreviations

Alt--Alternate Key
Caps Lock--Capital Letter Lock Key
Crtl--Control Key
Del--Delete Key
End--End Key
Esc--Escape Key
Home--Home Key
Ins--Insert Key
Num Lock--Number Lock Key
PgDn--Page Down Key
PgUp--Page Up Key
PrtSc--Print Screen Key
Return--Enter Command Key

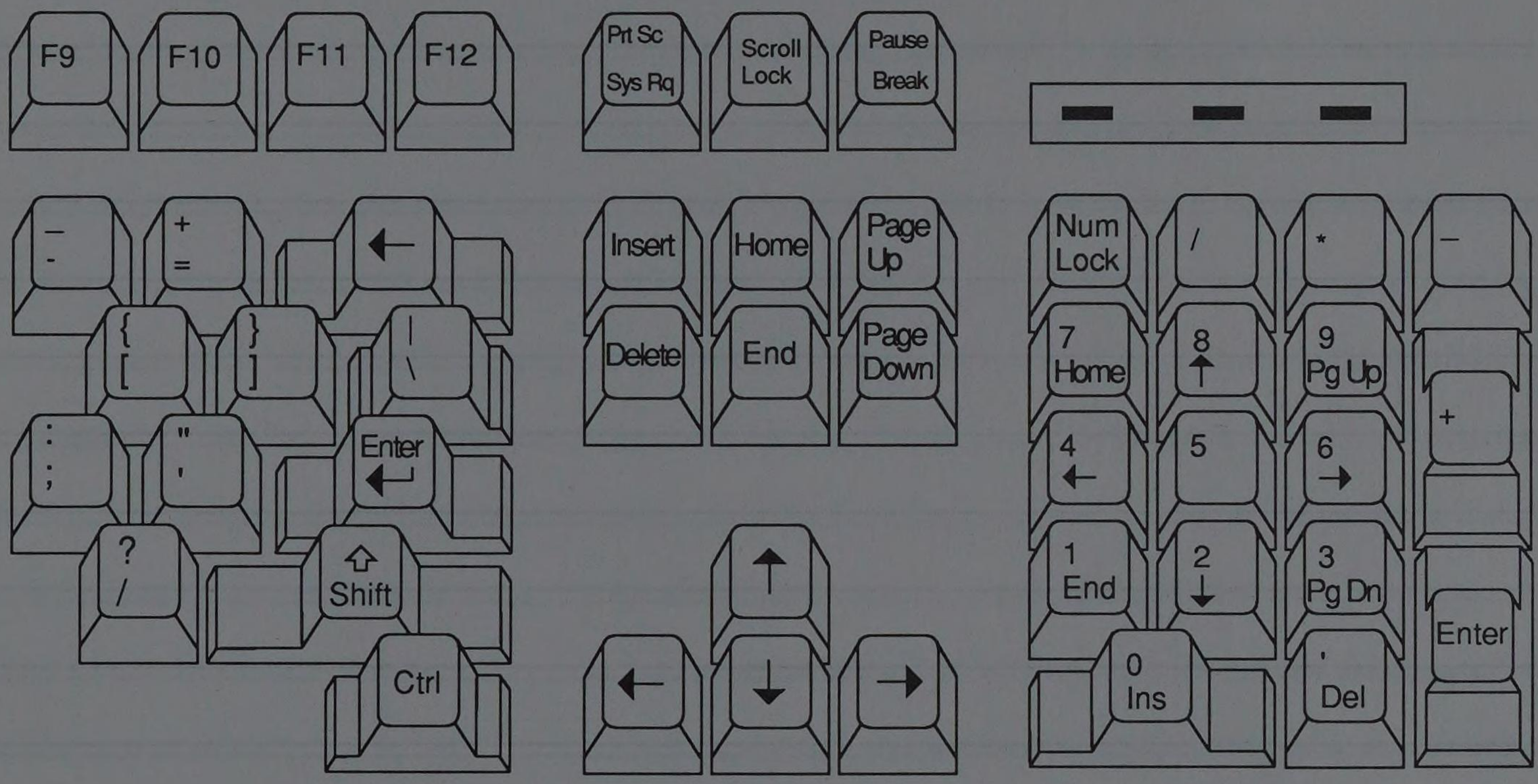


Enhanced Keyboard

DOS Keyboard

The last command entered at the DOS prompt is stored in a template buffer.

DEL	Does not copy the current template position to the command line
ESC	Voids the new command line and leaves template unchanged
F1	Plays back one character at a time from the template
F2	Plays back all characters from the template up to a specified character
F3	Plays back the remaining characters in the template
F4	Skips over all characters in the template up to a specified character
F5	Causes the current line to be the new template
F6	Inserts an End-of-file mark at the cursor location
Ins	Causes command line to toggle between insert mode (from keyboard) and insert mode (from the template)



IBM PC Abbreviations

Alt--Alternate Key
Caps Lock--Capital Letter Lock Key
Crtl--Control Key
Del--Delete Key
End--End Key
Esc--Escape Key
Home--Home Key
Ins--Insert Key
Num Lock--Number Lock Key
PgDn--Page Down Key
PgUp--Page Up Key
PrtSc--Print Screen Key
Enter--Enter Command Key

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

QUICK REFERENCE

MS-DOS®/PC DOS COMMANDS

APPEND C:\WORD\WP	Opens a path to data files in the C:\WORD\WP directory.
ATTRIB +A Filename	Turns the specified file's archive bit on.
ATTRIB +R Filename	Marks the specified file as read-only.
BACKUP C:*.* A:/S	Backs up all files from all directories and subdirectories on hard-disk C: to floppy disks in the A: drive.
CD\DOS	Switches the directory path to the \DOS subdirectory.
CHKDSK A:	Provides a status report on the internal working memory and disk storage contained on the diskette in drive A:.
CLS	Clears the monitor screen and moves the DOS prompt and cursor to the home position (upper left-hand corner).
COMP A:*.* C:	Compares all files on the disk in drive A: with like-named files on drive C:.
COPY A:FILE.DOC B:	Copies the file FILE.DOC from the disk in drive A: to the disk in drive B:
DATE	Prompts you to set or change the system date.
DEL A:*.*	Deletes all files from the disk in drive A:.
DIR	Lists files on the current disk drive including file names, date and time created, size in bytes, total number of files, and total bytes free on the disk.
DIR/P	Causes the screen to stop scrolling after 23 files on the current drive are viewed.
DIR/W	Displays a horizontal directory of files on the current drive. Lists only file names, total number of files, and bytes free on the disk.
DIR|SORT	Generates an alphabetical listing of all files on the default drive.
DISKCOMP A: B:	Compares the disk in drive A: to the disk in drive B:. The comparison is made track by track and sector by sector.
DISKCOPY A: B:	Makes an exact duplicate in drive B: of the original (source) disk in drive A:. The copy is made track by track and sector by sector.
EDLIN NEWFILE.BAT	Creates or edits a batch file named NEWFILE.BAT.

Continued on back side

FORMAT A:	Prepares a disk in drive A: to store data or programs. System files are not copied to the disk.
FORMAT A:/S	Prepares a self-booting system diskette to store data or programs.
MD DOS	Makes a new subdirectory named DOS under the current directory.
MODE LPT1:=COM1:	Redirects output from parallel port one to serial port one.
PATH C:\DOS	Opens a path to executable files in the C:\DOS subdirectory.
PROMPT PG	Changes the DOS prompt to display the current path, drive, and concludes the DOS prompt with the greater-than sign.
RD TRASH	Removes a subdirectory named TRASH that is under the current directory.
RD\TRASH	Removes a subdirectory named TRASH under the root directory.
RECOVER B:MESSEDUP.DOC	Recovers data from any good sectors storing portions of the file MESSEDUP.DOC on drive B:.
REN A:ONE.TXT TWO.TXT	Renames the file ONE.TXT (on the disk in drive A:) to TWO.TXT.
REPLACE *.* A:/A	Appends (copies) files from the source disk to the target disk if the files do not already exist on the target disk.
RESTORE A: C:*.*/S	Restores all files from the backup diskettes in drive A: to the appropriate directory or subdirectory on hard-disk C:.
RESTORE A: C:Filename	Restores the specified file from the backup disk in drive A: to the hard-disk C:.
SUBST E: C:\WORD\WP	Creates a virtual drive E:. Files in the C:\WORD\WP directory may be accessed as if they existed on drive E:.
SYS A:	Copies the two hidden system files to a disk in drive A:, thus making the disk bootable.
TIME	Prompts you to set or change the current time.
TREE	Lists the current directory and all its subdirectories.
TREE/F	Lists the current directory, all its associated files, plus all subdirectories under the current directory and their associated files.
XCOPY A: B:	Copies all files from drive A: to B:.
XCOPY A:*.DOC B:	Copies all document files (with the .DOC extension) from A: to B:.

THE MICROCOMPUTING SERIES

Understanding and Using Microcomputers, 2nd Edition
Understanding and Using Microcomputers, Alternate 2nd Edition

Introductory Books

Understanding and Using AppleWorks (Including AppleWorks 2.0)

Understanding and Using dBASE III (Including dBASE II)

Understanding and Using dBASE III PLUS

Understanding and Using dBASE IV

Understanding and Using DisplayWrite 4

Understanding and Using Educate-Ability

Understanding and Using FRAMEWORK

Understanding and Using Lotus 1–2–3

Understanding and Using Lotus 1–2–3 Release 2

Understanding and Using Lotus 1–2–3 Release 3

Understanding and Using Microsoft BASIC/IBM PC BASIC

Understanding and Using MS–DOS/PC DOS: The First Steps, 2nd Edition

Understanding and Using MS–DOS/PC DOS: A Complete Guide

Understanding and Using Microsoft Word

Understanding and Using MultiMate

Understanding and Using MS–DOS/PC DOS 4.0

Understanding and Using pfs:FILE/ REPORT

Understanding and Using PC–Write

Understanding and Using PFS:First Choice

Understanding and Using pfs:WRITE

Understanding and Using R:BASE 5000 (Including R:BASE SYSTEM V)

Understanding and Using SHAREWARE Application Software

Understanding and Using SuperCalc 3

Understanding and Using SuperCalc 4

Understanding and Using Symphony

Understanding and Using VP-Planner Plus

Understanding and Using WordPerfect

Understanding and Using WordPerfect 5.0

Understanding and Using WordStar

Understanding and Using WordStar 4.0

Understanding and Using Application Software

Developing and Using Office Applications with AppleWorks: Versions 1.0 through 2.0

Advanced Books

Developing and Using Decision Support Applications

Developing and Using Microcomputer Business Systems

ISBN 0-314-66573-0